C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatAnioMes.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatAnioMes;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatAnioMes;  
import mx.mbi.dgc.jpa.entity.CatAnioMes;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatAnioMes implements IfaceCatAnioMes {  
 @Autowired  
 private DAOCatAnioMes daoCatAnioMes;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatAnioMes> lista = new ArrayList<>();  
 List<CatAnioMes> objs = daoCatAnioMes.findAll();  
 for(CatAnioMes entity: objs) {  
 DominioCatAnioMes dom = new DominioCatAnioMes();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatAnioMes.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatAnioMes dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatAnioMes> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatAnioMes> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatAnioMes.findByAnioMesId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatAnioMes.findByAnioMesContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatAnioMes.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatAnioMes entity: objs) {  
 DominioCatAnioMes to = new DominioCatAnioMes();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatAnioMes dom = (DominioCatAnioMes)obj;  
 CatAnioMes entity = new CatAnioMes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatAnioMes.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatAnioMes.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatAnioMes dom = (DominioCatAnioMes)obj;  
 CatAnioMes entity = new CatAnioMes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatAnioMes.findById(entity.getAnioMesId()).  
 orElse(new CatAnioMes());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatAnioMes.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatAnioMes.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatAnioMes dom = (DominioCatAnioMes)obj;  
 CatAnioMes entity = new CatAnioMes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatAnioMes.findById(entity.getAnioMesId()).  
 orElse(new CatAnioMes());  
 daoCatAnioMes.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatAnioMes.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatAnioMes dom = (DominioCatAnioMes)obj;  
 CatAnioMes entity = new CatAnioMes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatAnioMes.findById(entity.getAnioMesId()).  
 orElse(new CatAnioMes());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatAnioMes.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatAnioMes.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatAnioMes.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatAnioMes existsById(Object obj) throws Exception {  
 try {  
 DominioCatAnioMes dom = (DominioCatAnioMes)obj;  
 CatAnioMes entity = new CatAnioMes();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatAnioMes.existsById(entity.getAnioMesId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatAnioMes.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatCapacidad.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatCapacidad;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatCapacidad;  
import mx.mbi.dgc.jpa.entity.CatCapacidad;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatCapacidad implements IfaceCatCapacidad {  
 @Autowired  
 private DAOCatCapacidad daoCatCapacidad;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatCapacidad> lista = new ArrayList<>();  
 List<CatCapacidad> objs = daoCatCapacidad.findAll();  
 for(CatCapacidad entity: objs) {  
 DominioCatCapacidad dom = new DominioCatCapacidad();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCapacidad.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatCapacidad dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatCapacidad> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatCapacidad> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatCapacidad.findByCapacidadId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatCapacidad.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatCapacidad.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatCapacidad entity: objs) {  
 DominioCatCapacidad to = new DominioCatCapacidad();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatCapacidad dom = (DominioCatCapacidad)obj;  
 CatCapacidad entity = new CatCapacidad();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCapacidad.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCapacidad.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatCapacidad dom = (DominioCatCapacidad)obj;  
 CatCapacidad entity = new CatCapacidad();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCapacidad.findById(entity.getCapacidadId()).  
 orElse(new CatCapacidad());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCapacidad.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCapacidad.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatCapacidad dom = (DominioCatCapacidad)obj;  
 CatCapacidad entity = new CatCapacidad();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCapacidad.findById(entity.getCapacidadId()).  
 orElse(new CatCapacidad());  
 daoCatCapacidad.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCapacidad.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatCapacidad dom = (DominioCatCapacidad)obj;  
 CatCapacidad entity = new CatCapacidad();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCapacidad.findById(entity.getCapacidadId()).  
 orElse(new CatCapacidad());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCapacidad.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatCapacidad.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCapacidad.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatCapacidad existsById(Object obj) throws Exception {  
 try {  
 DominioCatCapacidad dom = (DominioCatCapacidad)obj;  
 CatCapacidad entity = new CatCapacidad();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatCapacidad.existsById(entity.getCapacidadId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCapacidad.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatCargoFijo.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatCargoFijo;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatCargoFijo;  
import mx.mbi.dgc.jpa.entity.CatCargoFijo;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatCargoFijo implements IfaceCatCargoFijo {  
 @Autowired  
 private DAOCatCargoFijo daoCatCargoFijo;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatCargoFijo> lista = new ArrayList<>();  
 List<CatCargoFijo> objs = daoCatCargoFijo.findAll();  
 for(CatCargoFijo entity: objs) {  
 DominioCatCargoFijo dom = new DominioCatCargoFijo();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCargoFijo.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatCargoFijo dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatCargoFijo> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatCargoFijo> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatCargoFijo.findByCargoFijoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatCargoFijo.findByPathSoporteContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatCargoFijo.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatCargoFijo entity: objs) {  
 DominioCatCargoFijo to = new DominioCatCargoFijo();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatCargoFijo dom = (DominioCatCargoFijo)obj;  
 CatCargoFijo entity = new CatCargoFijo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCargoFijo.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCargoFijo.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatCargoFijo dom = (DominioCatCargoFijo)obj;  
 CatCargoFijo entity = new CatCargoFijo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCargoFijo.findById(entity.getCargoFijoId()).  
 orElse(new CatCargoFijo());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCargoFijo.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCargoFijo.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatCargoFijo dom = (DominioCatCargoFijo)obj;  
 CatCargoFijo entity = new CatCargoFijo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCargoFijo.findById(entity.getCargoFijoId()).  
 orElse(new CatCargoFijo());  
 daoCatCargoFijo.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCargoFijo.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatCargoFijo dom = (DominioCatCargoFijo)obj;  
 CatCargoFijo entity = new CatCargoFijo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCargoFijo.findById(entity.getCargoFijoId()).  
 orElse(new CatCargoFijo());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCargoFijo.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatCargoFijo.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCargoFijo.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatCargoFijo existsById(Object obj) throws Exception {  
 try {  
 DominioCatCargoFijo dom = (DominioCatCargoFijo)obj;  
 CatCargoFijo entity = new CatCargoFijo();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatCargoFijo.existsById(entity.getCargoFijoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCargoFijo.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatCargos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatCargos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatCargos;  
import mx.mbi.dgc.jpa.entity.CatCargos;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatCargos implements IfaceCatCargos {  
 @Autowired  
 private DAOCatCargos daoCatCargos;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatCargos> lista = new ArrayList<>();  
 List<CatCargos> objs = daoCatCargos.findAll();  
 for(CatCargos entity: objs) {  
 DominioCatCargos dom = new DominioCatCargos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCargos.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatCargos dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatCargos> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatCargos> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatCargos.findByCargoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatCargos.findByCargoContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatCargos.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatCargos entity: objs) {  
 DominioCatCargos to = new DominioCatCargos();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatCargos dom = (DominioCatCargos)obj;  
 CatCargos entity = new CatCargos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCargos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCargos.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatCargos dom = (DominioCatCargos)obj;  
 CatCargos entity = new CatCargos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCargos.findById(entity.getCargoId()).  
 orElse(new CatCargos());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCargos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCargos.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatCargos dom = (DominioCatCargos)obj;  
 CatCargos entity = new CatCargos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCargos.findById(entity.getCargoId()).  
 orElse(new CatCargos());  
 daoCatCargos.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCargos.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatCargos dom = (DominioCatCargos)obj;  
 CatCargos entity = new CatCargos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCargos.findById(entity.getCargoId()).  
 orElse(new CatCargos());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCargos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatCargos.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCargos.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatCargos existsById(Object obj) throws Exception {  
 try {  
 DominioCatCargos dom = (DominioCatCargos)obj;  
 CatCargos entity = new CatCargos();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatCargos.existsById(entity.getCargoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCargos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatClasEventos.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatClasificacionEventos;  
import mx.mbi.dgc.dominio.DominioCatClasificacionEventos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
import mx.mbi.dgc.jpa.entity.CatClasificacionEventos;  
  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service   
public class BusinessCatClasEventos implements IfaceCatClasEventos {  
 @Autowired  
 private DAOCatClasificacionEventos daoCatClasificacionEventos;  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatClasificacionEventos obj) throws Exception {  
   
 return null;  
 }  
  
 @Override  
 public List findAll() throws Exception {  
 List<CatClasificacionEventos> lista = new ArrayList<>();  
 lista = daoCatClasificacionEventos.findAll();  
 return lista;  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 return Collections.emptyList();  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {   
 DominioCatClasificacionEventos dominio = (DominioCatClasificacionEventos)obj;  
 CatClasificacionEventos entity = new CatClasificacionEventos();  
 JRUtil.copyProperties(entity, dominio);  
 entity = daoCatClasificacionEventos.save(entity);  
 JRUtil.copyProperties(dominio, entity);  
 dominio.setStatus(true);  
 dominio.setCode(200);  
 return dominio;  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 DominioCatClasificacionEventos dominio = (DominioCatClasificacionEventos)obj;  
 CatClasificacionEventos entity = new CatClasificacionEventos();  
 JRUtil.copyProperties(dominio, entity);  
 entity = daoCatClasificacionEventos.findById(entity.getClasificacionEventoId()).  
 orElse(new CatClasificacionEventos());  
 JRUtil.copyProperties(dominio, entity);  
 entity = daoCatClasificacionEventos.save(entity);  
 JRUtil.copyProperties(dominio, entity);  
 dominio.setStatus(true);  
 dominio.setCode(200);  
 return dominio;  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 DominioCatClasificacionEventos dominio = (DominioCatClasificacionEventos)obj;  
 CatClasificacionEventos entity = new CatClasificacionEventos();  
 JRUtil.copyProperties(dominio, entity);  
 entity = daoCatClasificacionEventos.findById(entity.getClasificacionEventoId()).  
 orElse(new CatClasificacionEventos());  
 daoCatClasificacionEventos.delete(entity);  
 dominio.setStatus(true);  
 dominio.setCode(200);  
 return dominio;  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 DominioCatClasificacionEventos dominio = (DominioCatClasificacionEventos)obj;  
 CatClasificacionEventos entity = new CatClasificacionEventos();  
 JRUtil.copyProperties(dominio,entity);  
 entity = daoCatClasificacionEventos.findById(entity.getClasificacionEventoId()).  
 orElse(new CatClasificacionEventos());  
 JRUtil.copyProperties(dominio, entity);  
 dominio.setStatus(true);  
 dominio.setCode(200);  
 return dominio;  
 }  
  
 @Override  
 public long count() throws Exception {  
 return daoCatClasificacionEventos.count();  
 }  
  
 @Override  
 public Object existsById(Object obj) throws Exception {  
 DominioCatClasificacionEventos dominio = new DominioCatClasificacionEventos();  
 CatClasificacionEventos entity = new CatClasificacionEventos();  
 JRUtil.copyProperties(dominio, entity);  
 dominio.setFound(daoCatClasificacionEventos.existsById(entity.getClasificacionEventoId()));  
 return dominio;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatClasificacion.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatClasificacion;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatClasificacion;  
import mx.mbi.dgc.jpa.entity.CatClasificacion;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatClasificacion implements IfaceCatClasificacion {  
 @Autowired  
 private DAOCatClasificacion daoCatClasificacion;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatClasificacion> lista = new ArrayList<>();  
 List<CatClasificacion> objs = daoCatClasificacion.findAll();  
 for(CatClasificacion entity: objs) {  
 DominioCatClasificacion dom = new DominioCatClasificacion();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacion.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatClasificacion dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatClasificacion> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatClasificacion> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatClasificacion.findByClasificacionId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatClasificacion.findByColorContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatClasificacion.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatClasificacion entity: objs) {  
 DominioCatClasificacion to = new DominioCatClasificacion();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatClasificacion dom = (DominioCatClasificacion)obj;  
 CatClasificacion entity = new CatClasificacion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClasificacion.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacion.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatClasificacion dom = (DominioCatClasificacion)obj;  
 CatClasificacion entity = new CatClasificacion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClasificacion.findById(entity.getClasificacionId()).  
 orElse(new CatClasificacion());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClasificacion.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacion.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatClasificacion dom = (DominioCatClasificacion)obj;  
 CatClasificacion entity = new CatClasificacion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClasificacion.findById(entity.getClasificacionId()).  
 orElse(new CatClasificacion());  
 daoCatClasificacion.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacion.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatClasificacion dom = (DominioCatClasificacion)obj;  
 CatClasificacion entity = new CatClasificacion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClasificacion.findById(entity.getClasificacionId()).  
 orElse(new CatClasificacion());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacion.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatClasificacion.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacion.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatClasificacion existsById(Object obj) throws Exception {  
 try {  
 DominioCatClasificacion dom = (DominioCatClasificacion)obj;  
 CatClasificacion entity = new CatClasificacion();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatClasificacion.existsById(entity.getClasificacionId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacion.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatClasificacionEvento.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatClasificacionEvento;  
import mx.mbi.dgc.dominio.DominioCatClasificacionEvento;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.jpa.entity.CatClasificacionEvento;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCatClasificacionEvento implements IfaceCatClasificacionEvento {  
 @Autowired  
 private DAOCatClasificacionEvento daoCatClasificacionEvento;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatClasificacionEvento> lista = new ArrayList<>();  
 List<CatClasificacionEvento> objs = daoCatClasificacionEvento.findAll();  
 for(CatClasificacionEvento entity: objs) {  
 DominioCatClasificacionEvento dom = new DominioCatClasificacionEvento();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacionEvento.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatClasificacionEvento dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatClasificacionEvento> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatClasificacionEvento> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatClasificacionEvento.findByClasificacionEventoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatClasificacionEvento.findByClasificacionEventoContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatClasificacionEvento.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatClasificacionEvento entity: objs) {  
 DominioCatClasificacionEvento to = new DominioCatClasificacionEvento();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 ex.printStackTrace();  
 throw new Exception("Error in BusinessCatClasificacionEvento.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatClasificacionEvento dom = (DominioCatClasificacionEvento)obj;  
 CatClasificacionEvento entity = new CatClasificacionEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClasificacionEvento.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacionEvento.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatClasificacionEvento dom = (DominioCatClasificacionEvento)obj;  
 CatClasificacionEvento entity = new CatClasificacionEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClasificacionEvento.findById(entity.getClasificacionEventoId()).  
 orElse(new CatClasificacionEvento());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClasificacionEvento.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacionEvento.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatClasificacionEvento dom = (DominioCatClasificacionEvento)obj;  
 CatClasificacionEvento entity = new CatClasificacionEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClasificacionEvento.findById(entity.getClasificacionEventoId()).  
 orElse(new CatClasificacionEvento());  
 daoCatClasificacionEvento.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacionEvento.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatClasificacionEvento dom = (DominioCatClasificacionEvento)obj;  
 CatClasificacionEvento entity = new CatClasificacionEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClasificacionEvento.findById(entity.getClasificacionEventoId()).  
 orElse(new CatClasificacionEvento());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacionEvento.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatClasificacionEvento.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacionEvento.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatClasificacionEvento existsById(Object obj) throws Exception {  
 try {  
 DominioCatClasificacionEvento dom = (DominioCatClasificacionEvento)obj;  
 CatClasificacionEvento entity = new CatClasificacionEvento();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatClasificacionEvento.existsById(entity.getClasificacionEventoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacionEvento.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatClasificacionEventos.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatClasificacionEventos;  
import mx.mbi.dgc.dominio.DominioCatClasificacionEventos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.jpa.entity.CatClasificacionEventos;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCatClasificacionEventos implements IfaceCatClasificacionEventos {  
 @Autowired  
 private DAOCatClasificacionEventos daoCatClasificacionEventos;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatClasificacionEventos> lista = new ArrayList<>();  
 List<CatClasificacionEventos> objs = daoCatClasificacionEventos.findAll();  
 for(CatClasificacionEventos entity: objs) {  
 DominioCatClasificacionEventos dom = new DominioCatClasificacionEventos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacionEventos.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatClasificacionEventos dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatClasificacionEventos> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatClasificacionEventos> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatClasificacionEventos.findByClasificacionEventoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatClasificacionEventos.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatClasificacionEventos.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatClasificacionEventos entity: objs) {  
 DominioCatClasificacionEventos to = new DominioCatClasificacionEventos();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatClasificacionEventos dom = (DominioCatClasificacionEventos)obj;  
 CatClasificacionEventos entity = new CatClasificacionEventos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClasificacionEventos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacionEventos.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatClasificacionEventos dom = (DominioCatClasificacionEventos)obj;  
 CatClasificacionEventos entity = new CatClasificacionEventos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClasificacionEventos.findById(entity.getClasificacionEventoId()).  
 orElse(new CatClasificacionEventos());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClasificacionEventos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacionEventos.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatClasificacionEventos dom = (DominioCatClasificacionEventos)obj;  
 CatClasificacionEventos entity = new CatClasificacionEventos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClasificacionEventos.findById(entity.getClasificacionEventoId()).  
 orElse(new CatClasificacionEventos());  
 daoCatClasificacionEventos.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacionEventos.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatClasificacionEventos dom = (DominioCatClasificacionEventos)obj;  
 CatClasificacionEventos entity = new CatClasificacionEventos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClasificacionEventos.findById(entity.getClasificacionEventoId()).  
 orElse(new CatClasificacionEventos());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacionEventos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatClasificacionEventos.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacionEventos.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatClasificacionEventos existsById(Object obj) throws Exception {  
 try {  
 DominioCatClasificacionEventos dom = (DominioCatClasificacionEventos)obj;  
 CatClasificacionEventos entity = new CatClasificacionEventos();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatClasificacionEventos.existsById(entity.getClasificacionEventoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClasificacionEventos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatClienteMonedas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatClienteMonedas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatClienteMonedas;  
import mx.mbi.dgc.jpa.entity.CatClienteMonedas;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatClienteMonedas implements IfaceCatClienteMonedas {  
 @Autowired  
 private DAOCatClienteMonedas daoCatClienteMonedas;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatClienteMonedas> lista = new ArrayList<>();  
 List<CatClienteMonedas> objs = daoCatClienteMonedas.findAll();  
 for(CatClienteMonedas entity: objs) {  
 DominioCatClienteMonedas dom = new DominioCatClienteMonedas();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClienteMonedas.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatClienteMonedas dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatClienteMonedas> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatClienteMonedas> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatClienteMonedas.findByClienteMonedaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatClienteMonedas.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatClienteMonedas.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatClienteMonedas entity: objs) {  
 DominioCatClienteMonedas to = new DominioCatClienteMonedas();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatClienteMonedas dom = (DominioCatClienteMonedas)obj;  
 CatClienteMonedas entity = new CatClienteMonedas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClienteMonedas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClienteMonedas.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatClienteMonedas dom = (DominioCatClienteMonedas)obj;  
 CatClienteMonedas entity = new CatClienteMonedas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClienteMonedas.findById(entity.getClienteMonedaId()).  
 orElse(new CatClienteMonedas());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClienteMonedas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClienteMonedas.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatClienteMonedas dom = (DominioCatClienteMonedas)obj;  
 CatClienteMonedas entity = new CatClienteMonedas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClienteMonedas.findById(entity.getClienteMonedaId()).  
 orElse(new CatClienteMonedas());  
 daoCatClienteMonedas.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClienteMonedas.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatClienteMonedas dom = (DominioCatClienteMonedas)obj;  
 CatClienteMonedas entity = new CatClienteMonedas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClienteMonedas.findById(entity.getClienteMonedaId()).  
 orElse(new CatClienteMonedas());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClienteMonedas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatClienteMonedas.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClienteMonedas.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatClienteMonedas existsById(Object obj) throws Exception {  
 try {  
 DominioCatClienteMonedas dom = (DominioCatClienteMonedas)obj;  
 CatClienteMonedas entity = new CatClienteMonedas();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatClienteMonedas.existsById(entity.getClienteMonedaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClienteMonedas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatClientes.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatClientes;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatClientes;  
import mx.mbi.dgc.jpa.entity.CatClientes;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatClientes implements IfaceCatClientes {  
 @Autowired  
 private DAOCatClientes daoCatClientes;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatClientes> lista = new ArrayList<>();  
 List<CatClientes> objs = daoCatClientes.findAll();  
 for(CatClientes entity: objs) {  
 DominioCatClientes dom = new DominioCatClientes();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClientes.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatClientes dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatClientes> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatClientes> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatClientes.findByClienteId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatClientes.findByNombreComercialContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatClientes.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatClientes entity: objs) {  
 DominioCatClientes to = new DominioCatClientes();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatClientes dom = (DominioCatClientes)obj;  
 CatClientes entity = new CatClientes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClientes.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClientes.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatClientes dom = (DominioCatClientes)obj;  
 CatClientes entity = new CatClientes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClientes.findById(entity.getClienteId()).  
 orElse(new CatClientes());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClientes.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClientes.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatClientes dom = (DominioCatClientes)obj;  
 CatClientes entity = new CatClientes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClientes.findById(entity.getClienteId()).  
 orElse(new CatClientes());  
 daoCatClientes.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClientes.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatClientes dom = (DominioCatClientes)obj;  
 CatClientes entity = new CatClientes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatClientes.findById(entity.getClienteId()).  
 orElse(new CatClientes());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClientes.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatClientes.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClientes.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatClientes existsById(Object obj) throws Exception {  
 try {  
 DominioCatClientes dom = (DominioCatClientes)obj;  
 CatClientes entity = new CatClientes();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatClientes.existsById(entity.getClienteId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatClientes.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatCombustible.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatCombustible;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatCombustible;  
import mx.mbi.dgc.jpa.entity.CatCombustible;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatCombustible implements IfaceCatCombustible {  
 @Autowired  
 private DAOCatCombustible daoCatCombustible;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatCombustible> lista = new ArrayList<>();  
 List<CatCombustible> objs = daoCatCombustible.findAll();  
 for(CatCombustible entity: objs) {  
 DominioCatCombustible dom = new DominioCatCombustible();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCombustible.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatCombustible dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatCombustible> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatCombustible> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatCombustible.findByCombustibleId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatCombustible.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatCombustible.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatCombustible entity: objs) {  
 DominioCatCombustible to = new DominioCatCombustible();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatCombustible dom = (DominioCatCombustible)obj;  
 CatCombustible entity = new CatCombustible();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCombustible.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCombustible.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatCombustible dom = (DominioCatCombustible)obj;  
 CatCombustible entity = new CatCombustible();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCombustible.findById(entity.getCombustibleId()).  
 orElse(new CatCombustible());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCombustible.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCombustible.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatCombustible dom = (DominioCatCombustible)obj;  
 CatCombustible entity = new CatCombustible();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCombustible.findById(entity.getCombustibleId()).  
 orElse(new CatCombustible());  
 daoCatCombustible.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCombustible.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatCombustible dom = (DominioCatCombustible)obj;  
 CatCombustible entity = new CatCombustible();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCombustible.findById(entity.getCombustibleId()).  
 orElse(new CatCombustible());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCombustible.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatCombustible.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCombustible.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatCombustible existsById(Object obj) throws Exception {  
 try {  
 DominioCatCombustible dom = (DominioCatCombustible)obj;  
 CatCombustible entity = new CatCombustible();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatCombustible.existsById(entity.getCombustibleId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCombustible.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatCompania.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatCompania;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatCompania;  
import mx.mbi.dgc.jpa.entity.CatCompania;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatCompania implements IfaceCatCompania {  
 @Autowired  
 private DAOCatCompania daoCatCompania;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatCompania> lista = new ArrayList<>();  
 List<CatCompania> objs = daoCatCompania.findAll();  
 for(CatCompania entity: objs) {  
 DominioCatCompania dom = new DominioCatCompania();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCompania.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatCompania dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatCompania> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatCompania> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatCompania.findByCompaniaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatCompania.findByNombreComercialContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatCompania.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatCompania entity: objs) {  
 DominioCatCompania to = new DominioCatCompania();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatCompania dom = (DominioCatCompania)obj;  
 CatCompania entity = new CatCompania();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCompania.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCompania.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatCompania dom = (DominioCatCompania)obj;  
 CatCompania entity = new CatCompania();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCompania.findById(entity.getCompaniaId()).  
 orElse(new CatCompania());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCompania.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCompania.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatCompania dom = (DominioCatCompania)obj;  
 CatCompania entity = new CatCompania();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCompania.findById(entity.getCompaniaId()).  
 orElse(new CatCompania());  
 daoCatCompania.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCompania.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatCompania dom = (DominioCatCompania)obj;  
 CatCompania entity = new CatCompania();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCompania.findById(entity.getCompaniaId()).  
 orElse(new CatCompania());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCompania.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatCompania.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCompania.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatCompania existsById(Object obj) throws Exception {  
 try {  
 DominioCatCompania dom = (DominioCatCompania)obj;  
 CatCompania entity = new CatCompania();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatCompania.existsById(entity.getCompaniaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCompania.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatCompaniaPlantas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatCompaniaPlantas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatCompaniaPlantas;  
import mx.mbi.dgc.jpa.entity.CatCompaniaPlantas;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatCompaniaPlantas implements IfaceCatCompaniaPlantas {  
 @Autowired  
 private DAOCatCompaniaPlantas daoCatCompaniaPlantas;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatCompaniaPlantas> lista = new ArrayList<>();  
 List<CatCompaniaPlantas> objs = daoCatCompaniaPlantas.findAll();  
 for(CatCompaniaPlantas entity: objs) {  
 DominioCatCompaniaPlantas dom = new DominioCatCompaniaPlantas();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCompaniaPlantas.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatCompaniaPlantas dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatCompaniaPlantas> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatCompaniaPlantas> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatCompaniaPlantas.findByCompaniaPlantaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatCompaniaPlantas.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatCompaniaPlantas.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatCompaniaPlantas entity: objs) {  
 DominioCatCompaniaPlantas to = new DominioCatCompaniaPlantas();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatCompaniaPlantas dom = (DominioCatCompaniaPlantas)obj;  
 CatCompaniaPlantas entity = new CatCompaniaPlantas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCompaniaPlantas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCompaniaPlantas.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatCompaniaPlantas dom = (DominioCatCompaniaPlantas)obj;  
 CatCompaniaPlantas entity = new CatCompaniaPlantas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCompaniaPlantas.findById(entity.getCompaniaPlantaId()).  
 orElse(new CatCompaniaPlantas());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCompaniaPlantas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCompaniaPlantas.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatCompaniaPlantas dom = (DominioCatCompaniaPlantas)obj;  
 CatCompaniaPlantas entity = new CatCompaniaPlantas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCompaniaPlantas.findById(entity.getCompaniaPlantaId()).  
 orElse(new CatCompaniaPlantas());  
 daoCatCompaniaPlantas.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCompaniaPlantas.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatCompaniaPlantas dom = (DominioCatCompaniaPlantas)obj;  
 CatCompaniaPlantas entity = new CatCompaniaPlantas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCompaniaPlantas.findById(entity.getCompaniaPlantaId()).  
 orElse(new CatCompaniaPlantas());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCompaniaPlantas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatCompaniaPlantas.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCompaniaPlantas.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatCompaniaPlantas existsById(Object obj) throws Exception {  
 try {  
 DominioCatCompaniaPlantas dom = (DominioCatCompaniaPlantas)obj;  
 CatCompaniaPlantas entity = new CatCompaniaPlantas();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatCompaniaPlantas.existsById(entity.getCompaniaPlantaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCompaniaPlantas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatComponentePeriodo.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatComponentePeriodo;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatComponentePeriodo;  
import mx.mbi.dgc.jpa.entity.CatComponentePeriodo;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatComponentePeriodo implements IfaceCatComponentePeriodo {  
 @Autowired  
 private DAOCatComponentePeriodo daoCatComponentePeriodo;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatComponentePeriodo> lista = new ArrayList<>();  
 List<CatComponentePeriodo> objs = daoCatComponentePeriodo.findAll();  
 for(CatComponentePeriodo entity: objs) {  
 DominioCatComponentePeriodo dom = new DominioCatComponentePeriodo();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatComponentePeriodo.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatComponentePeriodo dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatComponentePeriodo> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatComponentePeriodo> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatComponentePeriodo.findByComponentePeriodoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatComponentePeriodo.findByComponentePeriodoContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatComponentePeriodo.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatComponentePeriodo entity: objs) {  
 DominioCatComponentePeriodo to = new DominioCatComponentePeriodo();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatComponentePeriodo dom = (DominioCatComponentePeriodo)obj;  
 CatComponentePeriodo entity = new CatComponentePeriodo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatComponentePeriodo.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatComponentePeriodo.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatComponentePeriodo dom = (DominioCatComponentePeriodo)obj;  
 CatComponentePeriodo entity = new CatComponentePeriodo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatComponentePeriodo.findById(entity.getComponentePeriodoId()).  
 orElse(new CatComponentePeriodo());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatComponentePeriodo.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatComponentePeriodo.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatComponentePeriodo dom = (DominioCatComponentePeriodo)obj;  
 CatComponentePeriodo entity = new CatComponentePeriodo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatComponentePeriodo.findById(entity.getComponentePeriodoId()).  
 orElse(new CatComponentePeriodo());  
 daoCatComponentePeriodo.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatComponentePeriodo.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatComponentePeriodo dom = (DominioCatComponentePeriodo)obj;  
 CatComponentePeriodo entity = new CatComponentePeriodo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatComponentePeriodo.findById(entity.getComponentePeriodoId()).  
 orElse(new CatComponentePeriodo());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatComponentePeriodo.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatComponentePeriodo.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatComponentePeriodo.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatComponentePeriodo existsById(Object obj) throws Exception {  
 try {  
 DominioCatComponentePeriodo dom = (DominioCatComponentePeriodo)obj;  
 CatComponentePeriodo entity = new CatComponentePeriodo();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatComponentePeriodo.existsById(entity.getComponentePeriodoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatComponentePeriodo.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatComponentes.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatComponentes;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatComponentes;  
import mx.mbi.dgc.jpa.entity.CatComponentes;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatComponentes implements IfaceCatComponentes {  
 @Autowired  
 private DAOCatComponentes daoCatComponentes;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatComponentes> lista = new ArrayList<>();  
 List<CatComponentes> objs = daoCatComponentes.findAll();  
 for(CatComponentes entity: objs) {  
 DominioCatComponentes dom = new DominioCatComponentes();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatComponentes.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatComponentes dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatComponentes> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatComponentes> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatComponentes.findByComponenteId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatComponentes.findByComponenteContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatComponentes.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatComponentes entity: objs) {  
 DominioCatComponentes to = new DominioCatComponentes();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatComponentes dom = (DominioCatComponentes)obj;  
 CatComponentes entity = new CatComponentes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatComponentes.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatComponentes.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatComponentes dom = (DominioCatComponentes)obj;  
 CatComponentes entity = new CatComponentes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatComponentes.findById(entity.getComponenteId()).  
 orElse(new CatComponentes());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatComponentes.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatComponentes.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatComponentes dom = (DominioCatComponentes)obj;  
 CatComponentes entity = new CatComponentes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatComponentes.findById(entity.getComponenteId()).  
 orElse(new CatComponentes());  
 daoCatComponentes.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatComponentes.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatComponentes dom = (DominioCatComponentes)obj;  
 CatComponentes entity = new CatComponentes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatComponentes.findById(entity.getComponenteId()).  
 orElse(new CatComponentes());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatComponentes.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatComponentes.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatComponentes.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatComponentes existsById(Object obj) throws Exception {  
 try {  
 DominioCatComponentes dom = (DominioCatComponentes)obj;  
 CatComponentes entity = new CatComponentes();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatComponentes.existsById(entity.getComponenteId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatComponentes.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatCondiciones.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatCondiciones;  
import mx.mbi.dgc.dominio.DominioCatCondiciones;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.jpa.entity.CatCondiciones;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCatCondiciones implements IfaceCatCondiciones {  
 @Autowired  
 private DAOCatCondiciones daoCatCondiciones;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatCondiciones> lista = new ArrayList<>();  
 List<CatCondiciones> objs = daoCatCondiciones.findAll();  
 for(CatCondiciones entity: objs) {  
 DominioCatCondiciones dom = new DominioCatCondiciones();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCondiciones.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatCondiciones dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatCondiciones> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatCondiciones> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatCondiciones.findByCondicionId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatCondiciones.findByKeyCondicionContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatCondiciones.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatCondiciones entity: objs) {  
 DominioCatCondiciones to = new DominioCatCondiciones();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatCondiciones dom = (DominioCatCondiciones)obj;  
 CatCondiciones entity = new CatCondiciones();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCondiciones.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCondiciones.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatCondiciones dom = (DominioCatCondiciones)obj;  
 CatCondiciones entity = new CatCondiciones();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCondiciones.findById(entity.getCondicionId()).  
 orElse(new CatCondiciones());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCondiciones.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCondiciones.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatCondiciones dom = (DominioCatCondiciones)obj;  
 CatCondiciones entity = new CatCondiciones();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCondiciones.findById(entity.getCondicionId()).  
 orElse(new CatCondiciones());  
 daoCatCondiciones.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCondiciones.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatCondiciones dom = (DominioCatCondiciones)obj;  
 CatCondiciones entity = new CatCondiciones();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCondiciones.findById(entity.getCondicionId()).  
 orElse(new CatCondiciones());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCondiciones.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatCondiciones.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCondiciones.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatCondiciones existsById(Object obj) throws Exception {  
 try {  
 DominioCatCondiciones dom = (DominioCatCondiciones)obj;  
 CatCondiciones entity = new CatCondiciones();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatCondiciones.existsById(entity.getCondicionId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCondiciones.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatConexiones.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatConexiones;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatConexiones;  
import mx.mbi.dgc.jpa.entity.CatConexiones;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatConexiones implements IfaceCatConexiones {  
 @Autowired  
 private DAOCatConexiones daoCatConexiones;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatConexiones> lista = new ArrayList<>();  
 List<CatConexiones> objs = daoCatConexiones.findAll();  
 for(CatConexiones entity: objs) {  
 DominioCatConexiones dom = new DominioCatConexiones();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatConexiones.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatConexiones dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatConexiones> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatConexiones> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatConexiones.findByConexionId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatConexiones.findByConexionContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatConexiones.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatConexiones entity: objs) {  
 DominioCatConexiones to = new DominioCatConexiones();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatConexiones dom = (DominioCatConexiones)obj;  
 CatConexiones entity = new CatConexiones();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatConexiones.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatConexiones.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatConexiones dom = (DominioCatConexiones)obj;  
 CatConexiones entity = new CatConexiones();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatConexiones.findById(entity.getConexionId()).  
 orElse(new CatConexiones());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatConexiones.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatConexiones.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatConexiones dom = (DominioCatConexiones)obj;  
 CatConexiones entity = new CatConexiones();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatConexiones.findById(entity.getConexionId()).  
 orElse(new CatConexiones());  
 daoCatConexiones.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatConexiones.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatConexiones dom = (DominioCatConexiones)obj;  
 CatConexiones entity = new CatConexiones();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatConexiones.findById(entity.getConexionId()).  
 orElse(new CatConexiones());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatConexiones.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatConexiones.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatConexiones.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatConexiones existsById(Object obj) throws Exception {  
 try {  
 DominioCatConexiones dom = (DominioCatConexiones)obj;  
 CatConexiones entity = new CatConexiones();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatConexiones.existsById(entity.getConexionId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatConexiones.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatConfiguracionDetalle.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatConfiguracionDetalle;  
import mx.mbi.dgc.dominio.DominioCatConfiguracionDetalle;  
import mx.mbi.dgc.dominio.DominioCatConfiguracionEventos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.jpa.entity.CatConfiguracionDetalle;  
import mx.mbi.dgc.jpa.entity.CatConfiguracionEventos;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCatConfiguracionDetalle implements IfaceCatConfiguracionDetalle {  
 @Autowired  
 private DAOCatConfiguracionDetalle daoCatConfiguracionDetalle;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatConfiguracionDetalle> lista = new ArrayList<>();  
 List<CatConfiguracionDetalle> objs = daoCatConfiguracionDetalle.findAll();  
 for (CatConfiguracionDetalle entity : objs) {  
 DominioCatConfiguracionDetalle dom = new DominioCatConfiguracionDetalle();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionDetalle.findAll: " + ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatConfiguracionDetalle dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatConfiguracionDetalle> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatConfiguracionDetalle> objs = null;  
 if (dom.getSearch() != null && dom.getSearch().getValue() != null &&  
 ("" + dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs =  
 daoCatConfiguracionDetalle.findByConfiguracionDetalleId(new java.math.BigDecimal("" +  
 dom.getSearch()  
 .getValue()),  
 PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC :  
 Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex) {  
 try {  
 objs = null;  
 } catch (Exception ex2) {  
 }  
 }  
 } else {  
 objs =  
 daoCatConfiguracionDetalle.findAll(PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC : Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 }  
 for (CatConfiguracionDetalle entity : objs) {  
 DominioCatConfiguracionDetalle to = new DominioCatConfiguracionDetalle();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: " + ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatConfiguracionDetalle dom = (DominioCatConfiguracionDetalle) obj;  
 CatConfiguracionDetalle entity = new CatConfiguracionDetalle();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatConfiguracionDetalle.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionDetalle.save: " + ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatConfiguracionDetalle dom = (DominioCatConfiguracionDetalle) obj;  
 CatConfiguracionDetalle entity = new CatConfiguracionDetalle();  
 JRUtil.copyProperties(entity, dom);  
 entity =  
 daoCatConfiguracionDetalle.findById(entity.getConfiguracionDetalleId())  
 .orElse(new CatConfiguracionDetalle());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatConfiguracionDetalle.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionDetalle.update: " + ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatConfiguracionDetalle dom = (DominioCatConfiguracionDetalle) obj;  
 CatConfiguracionDetalle entity = new CatConfiguracionDetalle();  
 JRUtil.copyProperties(entity, dom);  
 entity =  
 daoCatConfiguracionDetalle.findById(entity.getConfiguracionDetalleId())  
 .orElse(new CatConfiguracionDetalle());  
 daoCatConfiguracionDetalle.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionDetalle.delete: " + ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatConfiguracionDetalle dom = (DominioCatConfiguracionDetalle) obj;  
 CatConfiguracionDetalle entity = new CatConfiguracionDetalle();  
 JRUtil.copyProperties(entity, dom);  
 entity =  
 daoCatConfiguracionDetalle.findById(entity.getConfiguracionDetalleId())  
 .orElse(new CatConfiguracionDetalle());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionDetalle.findById: " + ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatConfiguracionDetalle.count();  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionDetalle.count: " + ex);  
 }  
 }  
  
 @Override  
 public DominioCatConfiguracionDetalle existsById(Object obj) throws Exception {  
 try {  
 DominioCatConfiguracionDetalle dom = (DominioCatConfiguracionDetalle) obj;  
 CatConfiguracionDetalle entity = new CatConfiguracionDetalle();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatConfiguracionDetalle.existsById(entity.getConfiguracionDetalleId()));  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionDetalle.findById: " + ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
  
  
 @Override  
 public List<DominioCatConfiguracionDetalle> findByCatConfiguracionEvento(DominioCatConfiguracionEventos dom) throws Exception {  
 try {  
 List<DominioCatConfiguracionDetalle> lista = new ArrayList<>();  
 CatConfiguracionEventos entity = new CatConfiguracionEventos();  
 JRUtil.copyProperties(entity, dom);  
 List<CatConfiguracionDetalle> objs = daoCatConfiguracionDetalle.findByConfiguracionEventoId(entity);  
 for (CatConfiguracionDetalle entity2 : objs) {  
 DominioCatConfiguracionDetalle dom2 = new DominioCatConfiguracionDetalle();  
 JRUtil.copyProperties(dom2, entity2);  
 lista.add(dom2);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionEventos.findByCatConfiguracionEvento: " + ex);  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatConfiguracionEventos.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatConfiguracionEventos;  
import mx.mbi.dgc.dominio.DominioCatClasificacionEvento;  
import mx.mbi.dgc.dominio.DominioCatConfiguracionEventos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.jpa.entity.CatClasificacionEvento;  
import mx.mbi.dgc.jpa.entity.CatConfiguracionEventos;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCatConfiguracionEventos implements IfaceCatConfiguracionEventos {  
 @Autowired  
 private DAOCatConfiguracionEventos daoCatConfiguracionEventos;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatConfiguracionEventos> lista = new ArrayList<>();  
 List<CatConfiguracionEventos> objs = daoCatConfiguracionEventos.findAll();  
 for (CatConfiguracionEventos entity : objs) {  
 DominioCatConfiguracionEventos dom = new DominioCatConfiguracionEventos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionEventos.findAll: " + ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatConfiguracionEventos dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatConfiguracionEventos> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatConfiguracionEventos> objs = null;  
 if (dom.getSearch() != null && dom.getSearch().getValue() != null &&  
 ("" + dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs =  
 daoCatConfiguracionEventos.findByConfiguracionEventoId(new java.math.BigDecimal("" +  
 dom.getSearch()  
 .getValue()),  
 PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC :  
 Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex) {  
 try {  
 objs =  
 daoCatConfiguracionEventos.findByConfiguracionContaining("" + dom.getSearch().getValue(),  
 PageRequest.of(page,  
 dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC :  
 Sort.Direction.DESC,  
 dom.getCampos()  
 .get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex2) {  
 }  
 }  
 } else {  
 objs =  
 daoCatConfiguracionEventos.findAll(PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC : Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 }  
 for (CatConfiguracionEventos entity : objs) {  
 DominioCatConfiguracionEventos to = new DominioCatConfiguracionEventos();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: " + ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatConfiguracionEventos dom = (DominioCatConfiguracionEventos) obj;  
 CatConfiguracionEventos entity = new CatConfiguracionEventos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatConfiguracionEventos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionEventos.save: " + ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatConfiguracionEventos dom = (DominioCatConfiguracionEventos) obj;  
 CatConfiguracionEventos entity = new CatConfiguracionEventos();  
 JRUtil.copyProperties(entity, dom);  
 entity =  
 daoCatConfiguracionEventos.findById(entity.getConfiguracionEventoId())  
 .orElse(new CatConfiguracionEventos());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatConfiguracionEventos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionEventos.update: " + ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatConfiguracionEventos dom = (DominioCatConfiguracionEventos) obj;  
 CatConfiguracionEventos entity = new CatConfiguracionEventos();  
 JRUtil.copyProperties(entity, dom);  
 entity =  
 daoCatConfiguracionEventos.findById(entity.getConfiguracionEventoId())  
 .orElse(new CatConfiguracionEventos());  
 daoCatConfiguracionEventos.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionEventos.delete: " + ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatConfiguracionEventos dom = (DominioCatConfiguracionEventos) obj;  
 CatConfiguracionEventos entity = new CatConfiguracionEventos();  
 JRUtil.copyProperties(entity, dom);  
 entity =  
 daoCatConfiguracionEventos.findById(entity.getConfiguracionEventoId())  
 .orElse(new CatConfiguracionEventos());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionEventos.findById: " + ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatConfiguracionEventos.count();  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionEventos.count: " + ex);  
 }  
 }  
  
 @Override  
 public DominioCatConfiguracionEventos existsById(Object obj) throws Exception {  
 try {  
 DominioCatConfiguracionEventos dom = (DominioCatConfiguracionEventos) obj;  
 CatConfiguracionEventos entity = new CatConfiguracionEventos();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatConfiguracionEventos.existsById(entity.getConfiguracionEventoId()));  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionEventos.findById: " + ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
  
 @Override  
 public List<DominioCatConfiguracionEventos> findByCatClasificacionEvento(DominioCatClasificacionEvento dom) throws Exception {  
 try {  
 List<DominioCatConfiguracionEventos> lista = new ArrayList<>();  
 CatClasificacionEvento entity = new CatClasificacionEvento();  
 JRUtil.copyProperties(entity, dom);  
 List<CatConfiguracionEventos> objs = daoCatConfiguracionEventos.findByClasificacionEventoId(entity);  
 for (CatConfiguracionEventos entity2 : objs) {  
 DominioCatConfiguracionEventos dom2 = new DominioCatConfiguracionEventos();  
 JRUtil.copyProperties(dom2, entity2);  
 lista.add(dom2);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionEventos.findByCatClasificacionEvento: " + ex);  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatConfiguracionOpciones.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatConfiguracionOpciones;  
import mx.mbi.dgc.dao.DAOCatOpcionPlanta;  
import mx.mbi.dgc.dominio.DominioCatConfiguracionOpciones;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.negocio.bitacora.DominioPlantaOpcionesDetalle;  
import mx.mbi.dgc.jpa.entity.CatConfiguracionDetalle;  
import mx.mbi.dgc.jpa.entity.CatConfiguracionOpciones;  
import mx.mbi.dgc.jpa.entity.CatOpcionPlanta;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCatConfiguracionOpciones implements IfaceCatConfiguracionOpciones {  
 @Autowired  
 private DAOCatConfiguracionOpciones daoCatConfiguracionOpciones;  
 @Autowired  
 private DAOCatOpcionPlanta daoCatOpcionPlanta;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatConfiguracionOpciones> lista = new ArrayList<>();  
 List<CatConfiguracionOpciones> objs = daoCatConfiguracionOpciones.findAll();  
 for (CatConfiguracionOpciones entity : objs) {  
 DominioCatConfiguracionOpciones dom = new DominioCatConfiguracionOpciones();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionOpciones.findAll: " + ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatConfiguracionOpciones dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatConfiguracionOpciones> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatConfiguracionOpciones> objs = null;  
 if (dom.getSearch() != null && dom.getSearch().getValue() != null &&  
 ("" + dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs =  
 daoCatConfiguracionOpciones.findByConfiguracionOpcionesId(new java.math.BigDecimal("" +  
 dom.getSearch()  
 .getValue()),  
 PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC :  
 Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex) {  
 try {  
 objs = null;  
 } catch (Exception ex2) {  
 }  
 }  
 } else {  
 objs =  
 daoCatConfiguracionOpciones.findAll(PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC : Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 }  
 for (CatConfiguracionOpciones entity : objs) {  
 DominioCatConfiguracionOpciones to = new DominioCatConfiguracionOpciones();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: " + ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatConfiguracionOpciones dom = (DominioCatConfiguracionOpciones) obj;  
 CatConfiguracionOpciones entity = new CatConfiguracionOpciones();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatConfiguracionOpciones.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionOpciones.save: " + ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatConfiguracionOpciones dom = (DominioCatConfiguracionOpciones) obj;  
 CatConfiguracionOpciones entity = new CatConfiguracionOpciones();  
 JRUtil.copyProperties(entity, dom);  
 entity =  
 daoCatConfiguracionOpciones.findById(entity.getConfiguracionOpcionesId())  
 .orElse(new CatConfiguracionOpciones());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatConfiguracionOpciones.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionOpciones.update: " + ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatConfiguracionOpciones dom = (DominioCatConfiguracionOpciones) obj;  
 CatConfiguracionOpciones entity = new CatConfiguracionOpciones();  
 JRUtil.copyProperties(entity, dom);  
 entity =  
 daoCatConfiguracionOpciones.findById(entity.getConfiguracionOpcionesId())  
 .orElse(new CatConfiguracionOpciones());  
 daoCatConfiguracionOpciones.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionOpciones.delete: " + ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatConfiguracionOpciones dom = (DominioCatConfiguracionOpciones) obj;  
 CatConfiguracionOpciones entity = new CatConfiguracionOpciones();  
 JRUtil.copyProperties(entity, dom);  
 entity =  
 daoCatConfiguracionOpciones.findById(entity.getConfiguracionOpcionesId())  
 .orElse(new CatConfiguracionOpciones());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionOpciones.findById: " + ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatConfiguracionOpciones.count();  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionOpciones.count: " + ex);  
 }  
 }  
  
 @Override  
 public DominioCatConfiguracionOpciones existsById(Object obj) throws Exception {  
 try {  
 DominioCatConfiguracionOpciones dom = (DominioCatConfiguracionOpciones) obj;  
 CatConfiguracionOpciones entity = new CatConfiguracionOpciones();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatConfiguracionOpciones.existsById(entity.getConfiguracionOpcionesId()));  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionOpciones.findById: " + ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
  
 @Override  
 public List<DominioCatConfiguracionOpciones> findByCatConfiguracionDetalle(DominioPlantaOpcionesDetalle dom) throws Exception {  
 try {  
 List<DominioCatConfiguracionOpciones> lista = new ArrayList<>();  
 CatConfiguracionDetalle entity = new CatConfiguracionDetalle();  
 CatPlanta planta = new CatPlanta();  
 JRUtil.copyProperties(planta, dom.getDominioCatPlanta());  
 JRUtil.copyProperties(entity, dom.getDominioCatConfiguracionDetalle());  
 List<CatOpcionPlanta> opcionesPlanta = daoCatOpcionPlanta.findByPlantaId(planta);  
 List<CatConfiguracionOpciones> objs = daoCatConfiguracionOpciones.findByConfiguracionDetalleId(entity);  
 for (CatConfiguracionOpciones entity2 : objs) {  
 if (perteneceAPlanta(entity2, opcionesPlanta)) {  
 DominioCatConfiguracionOpciones dom2 = new DominioCatConfiguracionOpciones();  
 JRUtil.copyProperties(dom2, entity2);  
 lista.add(dom2);  
 }  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatConfiguracionEventos.findByCatConfiguracionDetalle: " + ex);  
 }  
 }  
  
 private boolean perteneceAPlanta(CatConfiguracionOpciones entity2, List<CatOpcionPlanta> opcionesPlanta) {  
 for (CatOpcionPlanta opcionPlanta : opcionesPlanta) {  
 if (entity2.getDetalleOpcionesId().getDetalleOpcionesId() ==  
 opcionPlanta.getDetalleOpcionesId().getDetalleOpcionesId())  
 return true;  
 }  
 return false;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatContratosAfectados.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatContratosAfectados;  
import mx.mbi.dgc.dominio.DominioCatContratosAfectados;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.jpa.entity.CatContratosAfectados;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCatContratosAfectados implements IfaceCatContratosAfectados {  
 @Autowired  
 private DAOCatContratosAfectados daoCatContratosAfectados;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatContratosAfectados> lista = new ArrayList<>();  
 List<CatContratosAfectados> objs = daoCatContratosAfectados.findAll();  
 for(CatContratosAfectados entity: objs) {  
 DominioCatContratosAfectados dom = new DominioCatContratosAfectados();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatContratosAfectados.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatContratosAfectados dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatContratosAfectados> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatContratosAfectados> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatContratosAfectados.findByContratoAfectadoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatContratosAfectados.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatContratosAfectados.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatContratosAfectados entity: objs) {  
 DominioCatContratosAfectados to = new DominioCatContratosAfectados();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatContratosAfectados dom = (DominioCatContratosAfectados)obj;  
 CatContratosAfectados entity = new CatContratosAfectados();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatContratosAfectados.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatContratosAfectados.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatContratosAfectados dom = (DominioCatContratosAfectados)obj;  
 CatContratosAfectados entity = new CatContratosAfectados();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatContratosAfectados.findById(entity.getContratoAfectadoId()).  
 orElse(new CatContratosAfectados());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatContratosAfectados.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatContratosAfectados.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatContratosAfectados dom = (DominioCatContratosAfectados)obj;  
 CatContratosAfectados entity = new CatContratosAfectados();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatContratosAfectados.findById(entity.getContratoAfectadoId()).  
 orElse(new CatContratosAfectados());  
 daoCatContratosAfectados.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatContratosAfectados.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatContratosAfectados dom = (DominioCatContratosAfectados)obj;  
 CatContratosAfectados entity = new CatContratosAfectados();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatContratosAfectados.findById(entity.getContratoAfectadoId()).  
 orElse(new CatContratosAfectados());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatContratosAfectados.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatContratosAfectados.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatContratosAfectados.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatContratosAfectados existsById(Object obj) throws Exception {  
 try {  
 DominioCatContratosAfectados dom = (DominioCatContratosAfectados)obj;  
 CatContratosAfectados entity = new CatContratosAfectados();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatContratosAfectados.existsById(entity.getContratoAfectadoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatContratosAfectados.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatCurvasCorreccion.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatCurvasCorreccion;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatCurvasCorreccion;  
import mx.mbi.dgc.jpa.entity.CatCurvasCorreccion;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatCurvasCorreccion implements IfaceCatCurvasCorreccion {  
 @Autowired  
 private DAOCatCurvasCorreccion daoCatCurvasCorreccion;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatCurvasCorreccion> lista = new ArrayList<>();  
 List<CatCurvasCorreccion> objs = daoCatCurvasCorreccion.findAll();  
 for(CatCurvasCorreccion entity: objs) {  
 DominioCatCurvasCorreccion dom = new DominioCatCurvasCorreccion();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCurvasCorreccion.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatCurvasCorreccion dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatCurvasCorreccion> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatCurvasCorreccion> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatCurvasCorreccion.findByCurvaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatCurvasCorreccion.findByFxContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatCurvasCorreccion.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatCurvasCorreccion entity: objs) {  
 DominioCatCurvasCorreccion to = new DominioCatCurvasCorreccion();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatCurvasCorreccion dom = (DominioCatCurvasCorreccion)obj;  
 CatCurvasCorreccion entity = new CatCurvasCorreccion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCurvasCorreccion.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCurvasCorreccion.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatCurvasCorreccion dom = (DominioCatCurvasCorreccion)obj;  
 CatCurvasCorreccion entity = new CatCurvasCorreccion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCurvasCorreccion.findById(entity.getCurvaId()).  
 orElse(new CatCurvasCorreccion());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCurvasCorreccion.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCurvasCorreccion.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatCurvasCorreccion dom = (DominioCatCurvasCorreccion)obj;  
 CatCurvasCorreccion entity = new CatCurvasCorreccion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCurvasCorreccion.findById(entity.getCurvaId()).  
 orElse(new CatCurvasCorreccion());  
 daoCatCurvasCorreccion.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCurvasCorreccion.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatCurvasCorreccion dom = (DominioCatCurvasCorreccion)obj;  
 CatCurvasCorreccion entity = new CatCurvasCorreccion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatCurvasCorreccion.findById(entity.getCurvaId()).  
 orElse(new CatCurvasCorreccion());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCurvasCorreccion.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatCurvasCorreccion.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCurvasCorreccion.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatCurvasCorreccion existsById(Object obj) throws Exception {  
 try {  
 DominioCatCurvasCorreccion dom = (DominioCatCurvasCorreccion)obj;  
 CatCurvasCorreccion entity = new CatCurvasCorreccion();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatCurvasCorreccion.existsById(entity.getCurvaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatCurvasCorreccion.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatDatosFiscales.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatDatosFiscales;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatDatosFiscales;  
import mx.mbi.dgc.jpa.entity.CatDatosFiscales;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatDatosFiscales implements IfaceCatDatosFiscales {  
 @Autowired  
 private DAOCatDatosFiscales daoCatDatosFiscales;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatDatosFiscales> lista = new ArrayList<>();  
 List<CatDatosFiscales> objs = daoCatDatosFiscales.findAll();  
 for(CatDatosFiscales entity: objs) {  
 DominioCatDatosFiscales dom = new DominioCatDatosFiscales();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatDatosFiscales.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatDatosFiscales dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatDatosFiscales> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatDatosFiscales> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatDatosFiscales.findByDatosFiscalesId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatDatosFiscales.findByCalleContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatDatosFiscales.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatDatosFiscales entity: objs) {  
 DominioCatDatosFiscales to = new DominioCatDatosFiscales();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatDatosFiscales dom = (DominioCatDatosFiscales)obj;  
 CatDatosFiscales entity = new CatDatosFiscales();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatDatosFiscales.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatDatosFiscales.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatDatosFiscales dom = (DominioCatDatosFiscales)obj;  
 CatDatosFiscales entity = new CatDatosFiscales();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatDatosFiscales.findById(entity.getDatosFiscalesId()).  
 orElse(new CatDatosFiscales());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatDatosFiscales.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatDatosFiscales.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatDatosFiscales dom = (DominioCatDatosFiscales)obj;  
 CatDatosFiscales entity = new CatDatosFiscales();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatDatosFiscales.findById(entity.getDatosFiscalesId()).  
 orElse(new CatDatosFiscales());  
 daoCatDatosFiscales.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatDatosFiscales.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatDatosFiscales dom = (DominioCatDatosFiscales)obj;  
 CatDatosFiscales entity = new CatDatosFiscales();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatDatosFiscales.findById(entity.getDatosFiscalesId()).  
 orElse(new CatDatosFiscales());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatDatosFiscales.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatDatosFiscales.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatDatosFiscales.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatDatosFiscales existsById(Object obj) throws Exception {  
 try {  
 DominioCatDatosFiscales dom = (DominioCatDatosFiscales)obj;  
 CatDatosFiscales entity = new CatDatosFiscales();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatDatosFiscales.existsById(entity.getDatosFiscalesId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatDatosFiscales.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatDetalleOpciones.java

package mx.mbi.dgc.business;  
  
import java.math.BigDecimal;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatDetalleOpciones;  
import mx.mbi.dgc.dao.DAOCatDetalles;  
import mx.mbi.dgc.dao.DAOCatOpcionPlanta;  
import mx.mbi.dgc.dominio.DominioCatDetalleOpciones;  
import mx.mbi.dgc.dominio.DominioCatDetalles;  
import mx.mbi.dgc.dominio.DominioCatOpcionPlanta;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.jpa.entity.CatDetalleOpciones;  
import mx.mbi.dgc.jpa.entity.CatDetalles;  
import mx.mbi.dgc.jpa.entity.CatOpcionPlanta;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCatDetalleOpciones implements IfaceCatDetalleOpciones {  
 @Autowired  
 private DAOCatDetalleOpciones daoCatDetalleOpciones;  
   
 @Autowired  
 private DAOCatOpcionPlanta daoCatOpcionPlanta;  
   
 @Autowired  
 private DAOCatDetalles daoCatDetalles;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatDetalleOpciones> lista = new ArrayList<>();  
 List<CatDetalleOpciones> objs = daoCatDetalleOpciones.findAll();  
 for (CatDetalleOpciones entity : objs) {  
 DominioCatDetalleOpciones dom = new DominioCatDetalleOpciones();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatDetalleOpciones.findAll: " + ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatDetalleOpciones dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatDetalleOpciones> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatDetalleOpciones> objs = null;  
 if (dom.getSearch() != null && dom.getSearch().getValue() != null &&  
 ("" + dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs =  
 daoCatDetalleOpciones.findByDetalleOpcionesId(new java.math.BigDecimal("" +  
 dom.getSearch()  
 .getValue()),  
 PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC :  
 Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex) {  
 try {  
 objs =  
 daoCatDetalleOpciones.findByOpcionContaining("" + dom.getSearch().getValue(),  
 PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC :  
 Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex2) {  
 }  
 }  
 } else {  
 objs =  
 daoCatDetalleOpciones.findAll(PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC : Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 }  
 for (CatDetalleOpciones entity : objs) {  
 DominioCatDetalleOpciones to = new DominioCatDetalleOpciones();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: " + ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatDetalleOpciones dom = (DominioCatDetalleOpciones) obj;  
 CatDetalleOpciones entity = new CatDetalleOpciones();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatDetalleOpciones.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatDetalleOpciones.save: " + ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatDetalleOpciones dom = (DominioCatDetalleOpciones) obj;  
 CatDetalleOpciones entity = new CatDetalleOpciones();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatDetalleOpciones.findById(entity.getDetalleOpcionesId()).orElse(new CatDetalleOpciones());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatDetalleOpciones.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatDetalleOpciones.update: " + ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatDetalleOpciones dom = (DominioCatDetalleOpciones) obj;  
 CatDetalleOpciones entity = new CatDetalleOpciones();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatDetalleOpciones.findById(entity.getDetalleOpcionesId()).orElse(new CatDetalleOpciones());  
 daoCatDetalleOpciones.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return obj;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatDetalleOpciones.delete: " + ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatDetalleOpciones dom = (DominioCatDetalleOpciones) obj;  
 CatDetalleOpciones entity = new CatDetalleOpciones();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatDetalleOpciones.findById(entity.getDetalleOpcionesId()).orElse(new CatDetalleOpciones());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatDetalleOpciones.findById: " + ex);  
 }  
 }  
  
 @Override  
 public List<DominioCatDetalleOpciones> findByDetalleId(DominioCatDetalles dominioCatDetalles) throws Exception {  
 try {  
 CatDetalles entity = daoCatDetalles.findByDetalle(dominioCatDetalles.getDetalle());  
 List<DominioCatDetalleOpciones> lista = new ArrayList<>();  
 List<CatDetalleOpciones> objs = daoCatDetalleOpciones.findByDetalleId(entity);  
 for (CatDetalleOpciones entity2 : objs) {  
 DominioCatDetalleOpciones dom = new DominioCatDetalleOpciones();  
 JRUtil.copyProperties(dom, entity2);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 return null;  
 }  
 }  
  
 @Override  
 public List<DominioCatDetalleOpciones> findByDetalle(DominioCatDetalles dominioCatDetalles) throws Exception {  
 try {  
 CatDetalles entity = daoCatDetalles.findByDetalleId(dominioCatDetalles.getDetalleId());  
 List<DominioCatDetalleOpciones> lista = new ArrayList<>();  
 List<CatDetalleOpciones> objs = daoCatDetalleOpciones.findByDetalleId(entity);  
 for (CatDetalleOpciones entity2 : objs) {  
 DominioCatDetalleOpciones dom = new DominioCatDetalleOpciones();  
 JRUtil.copyProperties(dom, entity2);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 return null;  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatDetalleOpciones.count();  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatDetalleOpciones.count: " + ex);  
 }  
 }  
  
 @Override  
 public DominioCatDetalleOpciones existsById(Object obj) throws Exception {  
 try {  
 DominioCatDetalleOpciones dom = (DominioCatDetalleOpciones) obj;  
 CatDetalleOpciones entity = new CatDetalleOpciones();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatDetalleOpciones.existsById(entity.getDetalleOpcionesId()));  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatDetalleOpciones.findById: " + ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
  
 @Override  
 public List<DominioCatDetalleOpciones> findByDetallePlantaId(DominioCatOpcionPlanta dom) throws Exception {  
 try {  
 List<DominioCatDetalleOpciones> lista = new ArrayList<>();  
 CatPlanta catPlanta = new CatPlanta();  
 CatDetalles catDetalles = new CatDetalles();  
 JRUtil.copyProperties(catPlanta, dom.getPlantaId());  
 JRUtil.copyProperties(catDetalles, dom.getDetalleOpcionesId().getDetalleId());  
 System.out.println("planta:" + dom.getPlantaId().getPlantaId());  
 List<CatOpcionPlanta> listCatOpcionPlanta = daoCatOpcionPlanta.findByPlantaId(catPlanta);  
 List<CatDetalleOpciones> listCatDetalles = daoCatDetalleOpciones.findByDetalleId(catDetalles);  
 for (CatDetalleOpciones aux : listCatDetalles)  
 for(CatOpcionPlanta aux2 : listCatOpcionPlanta){  
 if(aux.getDetalleOpcionesId() == aux2.getDetalleOpcionesId().getDetalleOpcionesId()){  
 DominioCatDetalleOpciones lista2 = new DominioCatDetalleOpciones();  
 JRUtil.copyProperties(lista2, aux);  
 lista.add(lista2);  
 System.out.println("planta:" + aux2.getPlantaId().getPlantaId() + "opcionId por planta" +aux2.getDetalleOpcionesId().getDetalleOpcionesId()+ "opcionId por detalle" + aux.getDetalleOpcionesId());  
 }  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception ("Error in BusinessCatDetalleOpciones.findByDetallePlantaId: " + ex);  
 }  
 }  
  
 @Override  
 public List<DominioCatDetalleOpciones> findCheck(DominioCatOpcionPlanta dom) throws Exception {  
 try {  
 int[] opcionArreglo = {7,10,17,11,20};  
 List<DominioCatDetalleOpciones> lista = new ArrayList<>();  
 for(int i : opcionArreglo) {  
 dom.getDetalleOpcionesId().getDetalleId().setDetalleId(BigDecimal.valueOf(i));  
 CatPlanta catPlanta = new CatPlanta();  
 CatDetalles catDetalles = new CatDetalles();  
 JRUtil.copyProperties(catPlanta, dom.getPlantaId());  
 JRUtil.copyProperties(catDetalles, dom.getDetalleOpcionesId().getDetalleId());  
 System.out.println("planta:" + dom.getPlantaId().getPlantaId());  
 List<CatOpcionPlanta> listCatOpcionPlanta = daoCatOpcionPlanta.findByPlantaId(catPlanta);  
 List<CatDetalleOpciones> listCatDetalles = daoCatDetalleOpciones.findByDetalleId(catDetalles);  
 for (CatDetalleOpciones aux : listCatDetalles)  
 for(CatOpcionPlanta aux2 : listCatOpcionPlanta){  
 if(aux.getDetalleOpcionesId() == aux2.getDetalleOpcionesId().getDetalleOpcionesId()){  
 DominioCatDetalleOpciones lista2 = new DominioCatDetalleOpciones();  
 JRUtil.copyProperties(lista2, aux);  
 lista.add(lista2);  
 System.out.println("planta:" + aux2.getPlantaId().getPlantaId() + "opcionId por planta" +aux2.getDetalleOpcionesId().getDetalleOpcionesId()+ "opcionId por detalle" + aux.getDetalleOpcionesId());  
 }  
 }  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception ("Error in BusinessCatDetalleOpciones.findByDetallePlantaId: " + ex);  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatDetalles.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatDetalles;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatDetalles;  
import mx.mbi.dgc.jpa.entity.CatDetalles;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatDetalles implements IfaceCatDetalles {  
 @Autowired  
 private DAOCatDetalles daoCatDetalles;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatDetalles> lista = new ArrayList<>();  
 List<CatDetalles> objs = daoCatDetalles.findAll();  
 for(CatDetalles entity: objs) {  
 DominioCatDetalles dom = new DominioCatDetalles();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatDetalles.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatDetalles dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatDetalles> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatDetalles> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatDetalles.findByDetalleId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatDetalles.findByDetalleContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatDetalles.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatDetalles entity: objs) {  
 DominioCatDetalles to = new DominioCatDetalles();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatDetalles dom = (DominioCatDetalles)obj;  
 CatDetalles entity = new CatDetalles();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatDetalles.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatDetalles.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatDetalles dom = (DominioCatDetalles)obj;  
 CatDetalles entity = new CatDetalles();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatDetalles.findById(entity.getDetalleId()).  
 orElse(new CatDetalles());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatDetalles.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatDetalles.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatDetalles dom = (DominioCatDetalles)obj;  
 CatDetalles entity = new CatDetalles();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatDetalles.findById(entity.getDetalleId()).orElse(new CatDetalles());  
 daoCatDetalles.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatDetalles.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatDetalles dom = (DominioCatDetalles)obj;  
 CatDetalles entity = new CatDetalles();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatDetalles.findById(entity.getDetalleId()).  
 orElse(new CatDetalles());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatDetalles.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatDetalles.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatDetalles.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatDetalles existsById(Object obj) throws Exception {  
 try {  
 DominioCatDetalles dom = (DominioCatDetalles)obj;  
 CatDetalles entity = new CatDetalles();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatDetalles.existsById(entity.getDetalleId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatDetalles.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatEstados.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatEstados;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatEstados;  
import mx.mbi.dgc.jpa.entity.CatEstados;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatEstados implements IfaceCatEstados {  
 @Autowired  
 private DAOCatEstados daoCatEstados;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatEstados> lista = new ArrayList<>();  
 List<CatEstados> objs = daoCatEstados.findAll();  
 for(CatEstados entity: objs) {  
 DominioCatEstados dom = new DominioCatEstados();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstados.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatEstados dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatEstados> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatEstados> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatEstados.findByEstadoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatEstados.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatEstados.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatEstados entity: objs) {  
 DominioCatEstados to = new DominioCatEstados();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatEstados dom = (DominioCatEstados)obj;  
 CatEstados entity = new CatEstados();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstados.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstados.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatEstados dom = (DominioCatEstados)obj;  
 CatEstados entity = new CatEstados();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstados.findById(entity.getEstadoId()).  
 orElse(new CatEstados());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstados.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstados.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatEstados dom = (DominioCatEstados)obj;  
 CatEstados entity = new CatEstados();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstados.findById(entity.getEstadoId()).  
 orElse(new CatEstados());  
 daoCatEstados.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstados.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatEstados dom = (DominioCatEstados)obj;  
 CatEstados entity = new CatEstados();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstados.findById(entity.getEstadoId()).  
 orElse(new CatEstados());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstados.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatEstados.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstados.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatEstados existsById(Object obj) throws Exception {  
 try {  
 DominioCatEstados dom = (DominioCatEstados)obj;  
 CatEstados entity = new CatEstados();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatEstados.existsById(entity.getEstadoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstados.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatEstatusBitacora.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatEstatusBitacora;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatEstatusBitacora;  
import mx.mbi.dgc.jpa.entity.CatEstatusBitacora;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatEstatusBitacora implements IfaceCatEstatusBitacora {  
 @Autowired  
 private DAOCatEstatusBitacora daoCatEstatusBitacora;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatEstatusBitacora> lista = new ArrayList<>();  
 List<CatEstatusBitacora> objs = daoCatEstatusBitacora.findAll();  
 for(CatEstatusBitacora entity: objs) {  
 DominioCatEstatusBitacora dom = new DominioCatEstatusBitacora();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusBitacora.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatEstatusBitacora dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatEstatusBitacora> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatEstatusBitacora> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatEstatusBitacora.findByEstatusBitacoraId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatEstatusBitacora.findByEstatusContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatEstatusBitacora.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatEstatusBitacora entity: objs) {  
 DominioCatEstatusBitacora to = new DominioCatEstatusBitacora();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatEstatusBitacora dom = (DominioCatEstatusBitacora)obj;  
 CatEstatusBitacora entity = new CatEstatusBitacora();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusBitacora.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusBitacora.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatEstatusBitacora dom = (DominioCatEstatusBitacora)obj;  
 CatEstatusBitacora entity = new CatEstatusBitacora();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusBitacora.findById(entity.getEstatusBitacoraId()).  
 orElse(new CatEstatusBitacora());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusBitacora.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusBitacora.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatEstatusBitacora dom = (DominioCatEstatusBitacora)obj;  
 CatEstatusBitacora entity = new CatEstatusBitacora();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusBitacora.findById(entity.getEstatusBitacoraId()).  
 orElse(new CatEstatusBitacora());  
 daoCatEstatusBitacora.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusBitacora.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatEstatusBitacora dom = (DominioCatEstatusBitacora)obj;  
 CatEstatusBitacora entity = new CatEstatusBitacora();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusBitacora.findById(entity.getEstatusBitacoraId()).  
 orElse(new CatEstatusBitacora());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusBitacora.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatEstatusBitacora.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusBitacora.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatEstatusBitacora existsById(Object obj) throws Exception {  
 try {  
 DominioCatEstatusBitacora dom = (DominioCatEstatusBitacora)obj;  
 CatEstatusBitacora entity = new CatEstatusBitacora();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatEstatusBitacora.existsById(entity.getEstatusBitacoraId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusBitacora.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatEstatusFactura.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatEstatusFactura;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatEstatusFactura;  
import mx.mbi.dgc.jpa.entity.CatEstatusFactura;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatEstatusFactura implements IfaceCatEstatusFactura {  
 @Autowired  
 private DAOCatEstatusFactura daoCatEstatusFactura;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatEstatusFactura> lista = new ArrayList<>();  
 List<CatEstatusFactura> objs = daoCatEstatusFactura.findAll();  
 for(CatEstatusFactura entity: objs) {  
 DominioCatEstatusFactura dom = new DominioCatEstatusFactura();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusFactura.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatEstatusFactura dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatEstatusFactura> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatEstatusFactura> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatEstatusFactura.findByEstatusFacturaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatEstatusFactura.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatEstatusFactura.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatEstatusFactura entity: objs) {  
 DominioCatEstatusFactura to = new DominioCatEstatusFactura();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatEstatusFactura dom = (DominioCatEstatusFactura)obj;  
 CatEstatusFactura entity = new CatEstatusFactura();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusFactura.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusFactura.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatEstatusFactura dom = (DominioCatEstatusFactura)obj;  
 CatEstatusFactura entity = new CatEstatusFactura();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusFactura.findById(entity.getEstatusFacturaId()).  
 orElse(new CatEstatusFactura());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusFactura.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusFactura.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatEstatusFactura dom = (DominioCatEstatusFactura)obj;  
 CatEstatusFactura entity = new CatEstatusFactura();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusFactura.findById(entity.getEstatusFacturaId()).  
 orElse(new CatEstatusFactura());  
 daoCatEstatusFactura.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusFactura.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatEstatusFactura dom = (DominioCatEstatusFactura)obj;  
 CatEstatusFactura entity = new CatEstatusFactura();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusFactura.findById(entity.getEstatusFacturaId()).  
 orElse(new CatEstatusFactura());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusFactura.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatEstatusFactura.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusFactura.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatEstatusFactura existsById(Object obj) throws Exception {  
 try {  
 DominioCatEstatusFactura dom = (DominioCatEstatusFactura)obj;  
 CatEstatusFactura entity = new CatEstatusFactura();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatEstatusFactura.existsById(entity.getEstatusFacturaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusFactura.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatEstatusPagos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatEstatusPagos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatEstatusPagos;  
import mx.mbi.dgc.jpa.entity.CatEstatusPagos;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatEstatusPagos implements IfaceCatEstatusPagos {  
 @Autowired  
 private DAOCatEstatusPagos daoCatEstatusPagos;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatEstatusPagos> lista = new ArrayList<>();  
 List<CatEstatusPagos> objs = daoCatEstatusPagos.findAll();  
 for(CatEstatusPagos entity: objs) {  
 DominioCatEstatusPagos dom = new DominioCatEstatusPagos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusPagos.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatEstatusPagos dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatEstatusPagos> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatEstatusPagos> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatEstatusPagos.findByEstatusPagoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatEstatusPagos.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatEstatusPagos.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatEstatusPagos entity: objs) {  
 DominioCatEstatusPagos to = new DominioCatEstatusPagos();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatEstatusPagos dom = (DominioCatEstatusPagos)obj;  
 CatEstatusPagos entity = new CatEstatusPagos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusPagos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusPagos.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatEstatusPagos dom = (DominioCatEstatusPagos)obj;  
 CatEstatusPagos entity = new CatEstatusPagos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusPagos.findById(entity.getEstatusPagoId()).  
 orElse(new CatEstatusPagos());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusPagos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusPagos.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatEstatusPagos dom = (DominioCatEstatusPagos)obj;  
 CatEstatusPagos entity = new CatEstatusPagos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusPagos.findById(entity.getEstatusPagoId()).  
 orElse(new CatEstatusPagos());  
 daoCatEstatusPagos.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusPagos.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatEstatusPagos dom = (DominioCatEstatusPagos)obj;  
 CatEstatusPagos entity = new CatEstatusPagos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusPagos.findById(entity.getEstatusPagoId()).  
 orElse(new CatEstatusPagos());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusPagos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatEstatusPagos.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusPagos.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatEstatusPagos existsById(Object obj) throws Exception {  
 try {  
 DominioCatEstatusPagos dom = (DominioCatEstatusPagos)obj;  
 CatEstatusPagos entity = new CatEstatusPagos();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatEstatusPagos.existsById(entity.getEstatusPagoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusPagos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatEstatusPerfil.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatEstatusPerfil;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatEstatusPerfil;  
import mx.mbi.dgc.jpa.entity.CatEstatusPerfil;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatEstatusPerfil implements IfaceCatEstatusPerfil {  
 @Autowired  
 private DAOCatEstatusPerfil daoCatEstatusPerfil;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatEstatusPerfil> lista = new ArrayList<>();  
 List<CatEstatusPerfil> objs = daoCatEstatusPerfil.findAll();  
 for(CatEstatusPerfil entity: objs) {  
 DominioCatEstatusPerfil dom = new DominioCatEstatusPerfil();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusPerfil.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatEstatusPerfil dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatEstatusPerfil> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatEstatusPerfil> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatEstatusPerfil.findByEstatusPerfilId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatEstatusPerfil.findByEstatusPerfilContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatEstatusPerfil.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatEstatusPerfil entity: objs) {  
 DominioCatEstatusPerfil to = new DominioCatEstatusPerfil();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatEstatusPerfil dom = (DominioCatEstatusPerfil)obj;  
 CatEstatusPerfil entity = new CatEstatusPerfil();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusPerfil.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusPerfil.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatEstatusPerfil dom = (DominioCatEstatusPerfil)obj;  
 CatEstatusPerfil entity = new CatEstatusPerfil();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusPerfil.findById(entity.getEstatusPerfilId()).  
 orElse(new CatEstatusPerfil());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusPerfil.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusPerfil.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatEstatusPerfil dom = (DominioCatEstatusPerfil)obj;  
 CatEstatusPerfil entity = new CatEstatusPerfil();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusPerfil.findById(entity.getEstatusPerfilId()).  
 orElse(new CatEstatusPerfil());  
 daoCatEstatusPerfil.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusPerfil.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatEstatusPerfil dom = (DominioCatEstatusPerfil)obj;  
 CatEstatusPerfil entity = new CatEstatusPerfil();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusPerfil.findById(entity.getEstatusPerfilId()).  
 orElse(new CatEstatusPerfil());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusPerfil.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatEstatusPerfil.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusPerfil.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatEstatusPerfil existsById(Object obj) throws Exception {  
 try {  
 DominioCatEstatusPerfil dom = (DominioCatEstatusPerfil)obj;  
 CatEstatusPerfil entity = new CatEstatusPerfil();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatEstatusPerfil.existsById(entity.getEstatusPerfilId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusPerfil.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatEstatusVariable.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatEstatusVariable;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatEstatusVariable;  
import mx.mbi.dgc.jpa.entity.CatEstatusVariable;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatEstatusVariable implements IfaceCatEstatusVariable {  
 @Autowired  
 private DAOCatEstatusVariable daoCatEstatusVariable;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatEstatusVariable> lista = new ArrayList<>();  
 List<CatEstatusVariable> objs = daoCatEstatusVariable.findAll();  
 for(CatEstatusVariable entity: objs) {  
 DominioCatEstatusVariable dom = new DominioCatEstatusVariable();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusVariable.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatEstatusVariable dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatEstatusVariable> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatEstatusVariable> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatEstatusVariable.findByEstatusVarId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatEstatusVariable.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatEstatusVariable.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatEstatusVariable entity: objs) {  
 DominioCatEstatusVariable to = new DominioCatEstatusVariable();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatEstatusVariable dom = (DominioCatEstatusVariable)obj;  
 CatEstatusVariable entity = new CatEstatusVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusVariable.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusVariable.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatEstatusVariable dom = (DominioCatEstatusVariable)obj;  
 CatEstatusVariable entity = new CatEstatusVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusVariable.findById(entity.getEstatusVariableId()).  
 orElse(new CatEstatusVariable());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusVariable.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusVariable.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatEstatusVariable dom = (DominioCatEstatusVariable)obj;  
 CatEstatusVariable entity = new CatEstatusVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusVariable.findById(entity.getEstatusVariableId()).  
 orElse(new CatEstatusVariable());  
 daoCatEstatusVariable.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusVariable.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatEstatusVariable dom = (DominioCatEstatusVariable)obj;  
 CatEstatusVariable entity = new CatEstatusVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEstatusVariable.findById(entity.getEstatusVariableId()).  
 orElse(new CatEstatusVariable());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusVariable.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatEstatusVariable.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusVariable.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatEstatusVariable existsById(Object obj) throws Exception {  
 try {  
 DominioCatEstatusVariable dom = (DominioCatEstatusVariable)obj;  
 CatEstatusVariable entity = new CatEstatusVariable();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatEstatusVariable.existsById(entity.getEstatusVariableId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEstatusVariable.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatEtapas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatEtapas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatEtapas;  
import mx.mbi.dgc.jpa.entity.CatEtapas;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatEtapas implements IfaceCatEtapas {  
 @Autowired  
 private DAOCatEtapas daoCatEtapas;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatEtapas> lista = new ArrayList<>();  
 List<CatEtapas> objs = daoCatEtapas.findAll();  
 for(CatEtapas entity: objs) {  
 DominioCatEtapas dom = new DominioCatEtapas();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEtapas.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatEtapas dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatEtapas> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatEtapas> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatEtapas.findByEtapaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatEtapas.findByEtapaContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatEtapas.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatEtapas entity: objs) {  
 DominioCatEtapas to = new DominioCatEtapas();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatEtapas dom = (DominioCatEtapas)obj;  
 CatEtapas entity = new CatEtapas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEtapas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEtapas.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatEtapas dom = (DominioCatEtapas)obj;  
 CatEtapas entity = new CatEtapas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEtapas.findById(entity.getEtapaId()).  
 orElse(new CatEtapas());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEtapas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEtapas.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatEtapas dom = (DominioCatEtapas)obj;  
 CatEtapas entity = new CatEtapas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEtapas.findById(entity.getEtapaId()).  
 orElse(new CatEtapas());  
 daoCatEtapas.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEtapas.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatEtapas dom = (DominioCatEtapas)obj;  
 CatEtapas entity = new CatEtapas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEtapas.findById(entity.getEtapaId()).  
 orElse(new CatEtapas());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEtapas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatEtapas.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEtapas.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatEtapas existsById(Object obj) throws Exception {  
 try {  
 DominioCatEtapas dom = (DominioCatEtapas)obj;  
 CatEtapas entity = new CatEtapas();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatEtapas.existsById(entity.getEtapaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEtapas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatEventos.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatEventos;  
import mx.mbi.dgc.dominio.DominioCatEventos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.jpa.entity.CatEventos;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCatEventos implements IfaceCatEventos {  
 @Autowired  
 private DAOCatEventos daoCatEventos;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatEventos> lista = new ArrayList<>();  
 List<CatEventos> objs = daoCatEventos.findAll();  
 for(CatEventos entity: objs) {  
 DominioCatEventos dom = new DominioCatEventos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEventos.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatEventos dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatEventos> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatEventos> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatEventos.findByEventoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatEventos.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatEventos.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatEventos entity: objs) {  
 DominioCatEventos to = new DominioCatEventos();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatEventos dom = (DominioCatEventos)obj;  
 CatEventos entity = new CatEventos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEventos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEventos.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatEventos dom = (DominioCatEventos)obj;  
 CatEventos entity = new CatEventos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEventos.findById(entity.getEventoId()).  
 orElse(new CatEventos());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEventos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEventos.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatEventos dom = (DominioCatEventos)obj;  
 CatEventos entity = new CatEventos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEventos.findById(entity.getEventoId()).  
 orElse(new CatEventos());  
 daoCatEventos.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEventos.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatEventos dom = (DominioCatEventos)obj;  
 CatEventos entity = new CatEventos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatEventos.findById(entity.getEventoId()).  
 orElse(new CatEventos());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEventos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatEventos.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEventos.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatEventos existsById(Object obj) throws Exception {  
 try {  
 DominioCatEventos dom = (DominioCatEventos)obj;  
 CatEventos entity = new CatEventos();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatEventos.existsById(entity.getEventoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatEventos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatFactorTipo.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatFactorTipo;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatFactorTipo;  
import mx.mbi.dgc.jpa.entity.CatFactorTipo;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatFactorTipo implements IfaceCatFactorTipo {  
 @Autowired  
 private DAOCatFactorTipo daoCatFactorTipo;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatFactorTipo> lista = new ArrayList<>();  
 List<CatFactorTipo> objs = daoCatFactorTipo.findAll();  
 for(CatFactorTipo entity: objs) {  
 DominioCatFactorTipo dom = new DominioCatFactorTipo();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFactorTipo.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatFactorTipo dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatFactorTipo> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatFactorTipo> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatFactorTipo.findByFactorTipoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatFactorTipo.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatFactorTipo.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatFactorTipo entity: objs) {  
 DominioCatFactorTipo to = new DominioCatFactorTipo();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatFactorTipo dom = (DominioCatFactorTipo)obj;  
 CatFactorTipo entity = new CatFactorTipo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFactorTipo.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFactorTipo.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatFactorTipo dom = (DominioCatFactorTipo)obj;  
 CatFactorTipo entity = new CatFactorTipo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFactorTipo.findById(entity.getFactorTipoId()).  
 orElse(new CatFactorTipo());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFactorTipo.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFactorTipo.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatFactorTipo dom = (DominioCatFactorTipo)obj;  
 CatFactorTipo entity = new CatFactorTipo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFactorTipo.findById(entity.getFactorTipoId()).  
 orElse(new CatFactorTipo());  
 daoCatFactorTipo.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFactorTipo.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatFactorTipo dom = (DominioCatFactorTipo)obj;  
 CatFactorTipo entity = new CatFactorTipo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFactorTipo.findById(entity.getFactorTipoId()).  
 orElse(new CatFactorTipo());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFactorTipo.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatFactorTipo.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFactorTipo.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatFactorTipo existsById(Object obj) throws Exception {  
 try {  
 DominioCatFactorTipo dom = (DominioCatFactorTipo)obj;  
 CatFactorTipo entity = new CatFactorTipo();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatFactorTipo.existsById(entity.getFactorTipoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFactorTipo.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatFacturaProductos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatFacturaProductos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatFacturaProductos;  
import mx.mbi.dgc.jpa.entity.CatFacturaProductos;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatFacturaProductos implements IfaceCatFacturaProductos {  
 @Autowired  
 private DAOCatFacturaProductos daoCatFacturaProductos;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatFacturaProductos> lista = new ArrayList<>();  
 List<CatFacturaProductos> objs = daoCatFacturaProductos.findAll();  
 for(CatFacturaProductos entity: objs) {  
 DominioCatFacturaProductos dom = new DominioCatFacturaProductos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFacturaProductos.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatFacturaProductos dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatFacturaProductos> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatFacturaProductos> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatFacturaProductos.findByFacturaProductoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatFacturaProductos.findByObservacionesContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatFacturaProductos.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatFacturaProductos entity: objs) {  
 DominioCatFacturaProductos to = new DominioCatFacturaProductos();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatFacturaProductos dom = (DominioCatFacturaProductos)obj;  
 CatFacturaProductos entity = new CatFacturaProductos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFacturaProductos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFacturaProductos.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatFacturaProductos dom = (DominioCatFacturaProductos)obj;  
 CatFacturaProductos entity = new CatFacturaProductos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFacturaProductos.findById(entity.getFacturaProductoId()).  
 orElse(new CatFacturaProductos());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFacturaProductos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFacturaProductos.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatFacturaProductos dom = (DominioCatFacturaProductos)obj;  
 CatFacturaProductos entity = new CatFacturaProductos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFacturaProductos.findById(entity.getFacturaProductoId()).  
 orElse(new CatFacturaProductos());  
 daoCatFacturaProductos.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFacturaProductos.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatFacturaProductos dom = (DominioCatFacturaProductos)obj;  
 CatFacturaProductos entity = new CatFacturaProductos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFacturaProductos.findById(entity.getFacturaProductoId()).  
 orElse(new CatFacturaProductos());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFacturaProductos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatFacturaProductos.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFacturaProductos.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatFacturaProductos existsById(Object obj) throws Exception {  
 try {  
 DominioCatFacturaProductos dom = (DominioCatFacturaProductos)obj;  
 CatFacturaProductos entity = new CatFacturaProductos();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatFacturaProductos.existsById(entity.getFacturaProductoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFacturaProductos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatFirmaCincominutal.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatFirmaCincominutal;  
import mx.mbi.dgc.dominio.DominioCatFirmaCincominutal;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.jpa.entity.CatFirmaCincominutal;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCatFirmaCincominutal implements IfaceCatFirmaCincominutal {  
 @Autowired  
 private DAOCatFirmaCincominutal daoCatFirmaCincominutal;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatFirmaCincominutal> lista = new ArrayList<>();  
 List<CatFirmaCincominutal> objs = daoCatFirmaCincominutal.findAll();  
 for (CatFirmaCincominutal entity : objs) {  
 DominioCatFirmaCincominutal dom = new DominioCatFirmaCincominutal();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatFirmaCincominutal.findAll: " + ex);  
 }  
 }  
  
 @Override  
 public List<DominioCatFirmaCincominutal> findAllLimit() throws Exception {  
 try {  
 List<DominioCatFirmaCincominutal> lista = new ArrayList<>();  
 List<CatFirmaCincominutal> objs = daoCatFirmaCincominutal.findFirst288ByOrderByFirmaId();  
 for (CatFirmaCincominutal entity : objs) {  
 DominioCatFirmaCincominutal dom = new DominioCatFirmaCincominutal();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatFirmaCincominutal.findAll: " + ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatFirmaCincominutal dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatFirmaCincominutal> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatFirmaCincominutal> objs = null;  
 if (dom.getSearch() != null && dom.getSearch().getValue() != null &&  
 ("" + dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs =  
 daoCatFirmaCincominutal.findByFirmaId(new java.math.BigDecimal("" + dom.getSearch().getValue()),  
 PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC : Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex) {  
 try {  
 objs =  
 daoCatFirmaCincominutal.findByFirmaContaining("" + dom.getSearch().getValue(),  
 PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC :  
 Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex2) {  
 }  
 }  
 } else {  
 objs =  
 daoCatFirmaCincominutal.findAll(PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC : Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 }  
 for (CatFirmaCincominutal entity : objs) {  
 DominioCatFirmaCincominutal to = new DominioCatFirmaCincominutal();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: " + ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatFirmaCincominutal dom = (DominioCatFirmaCincominutal) obj;  
 CatFirmaCincominutal entity = new CatFirmaCincominutal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFirmaCincominutal.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatFirmaCincominutal.save: " + ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatFirmaCincominutal dom = (DominioCatFirmaCincominutal) obj;  
 CatFirmaCincominutal entity = new CatFirmaCincominutal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFirmaCincominutal.findById(entity.getFirmaId()).orElse(new CatFirmaCincominutal());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFirmaCincominutal.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatFirmaCincominutal.update: " + ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatFirmaCincominutal dom = (DominioCatFirmaCincominutal) obj;  
 CatFirmaCincominutal entity = new CatFirmaCincominutal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFirmaCincominutal.findById(entity.getFirmaId()).orElse(new CatFirmaCincominutal());  
 daoCatFirmaCincominutal.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatFirmaCincominutal.delete: " + ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatFirmaCincominutal dom = (DominioCatFirmaCincominutal) obj;  
 CatFirmaCincominutal entity = new CatFirmaCincominutal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFirmaCincominutal.findById(entity.getFirmaId()).orElse(new CatFirmaCincominutal());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatFirmaCincominutal.findById: " + ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatFirmaCincominutal.count();  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatFirmaCincominutal.count: " + ex);  
 }  
 }  
  
 @Override  
 public DominioCatFirmaCincominutal existsById(Object obj) throws Exception {  
 try {  
 DominioCatFirmaCincominutal dom = (DominioCatFirmaCincominutal) obj;  
 CatFirmaCincominutal entity = new CatFirmaCincominutal();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatFirmaCincominutal.existsById(entity.getFirmaId()));  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatFirmaCincominutal.findById: " + ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatFormasPago.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatFormasPago;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatFormasPago;  
import mx.mbi.dgc.jpa.entity.CatFormasPago;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatFormasPago implements IfaceCatFormasPago {  
 @Autowired  
 private DAOCatFormasPago daoCatFormasPago;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatFormasPago> lista = new ArrayList<>();  
 List<CatFormasPago> objs = daoCatFormasPago.findAll();  
 for(CatFormasPago entity: objs) {  
 DominioCatFormasPago dom = new DominioCatFormasPago();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormasPago.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatFormasPago dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatFormasPago> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatFormasPago> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatFormasPago.findByFormaPagoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatFormasPago.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatFormasPago.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatFormasPago entity: objs) {  
 DominioCatFormasPago to = new DominioCatFormasPago();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatFormasPago dom = (DominioCatFormasPago)obj;  
 CatFormasPago entity = new CatFormasPago();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFormasPago.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormasPago.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatFormasPago dom = (DominioCatFormasPago)obj;  
 CatFormasPago entity = new CatFormasPago();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFormasPago.findById(entity.getFormaPagoId()).  
 orElse(new CatFormasPago());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFormasPago.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormasPago.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatFormasPago dom = (DominioCatFormasPago)obj;  
 CatFormasPago entity = new CatFormasPago();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFormasPago.findById(entity.getFormaPagoId()).  
 orElse(new CatFormasPago());  
 daoCatFormasPago.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormasPago.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatFormasPago dom = (DominioCatFormasPago)obj;  
 CatFormasPago entity = new CatFormasPago();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFormasPago.findById(entity.getFormaPagoId()).  
 orElse(new CatFormasPago());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormasPago.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatFormasPago.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormasPago.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatFormasPago existsById(Object obj) throws Exception {  
 try {  
 DominioCatFormasPago dom = (DominioCatFormasPago)obj;  
 CatFormasPago entity = new CatFormasPago();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatFormasPago.existsById(entity.getFormaPagoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormasPago.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatFormulaComponentes.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatFormulaComponentes;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatFormulaComponentes;  
import mx.mbi.dgc.jpa.entity.CatFormulaComponentes;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatFormulaComponentes implements IfaceCatFormulaComponentes {  
 @Autowired  
 private DAOCatFormulaComponentes daoCatFormulaComponentes;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatFormulaComponentes> lista = new ArrayList<>();  
 List<CatFormulaComponentes> objs = daoCatFormulaComponentes.findAll();  
 for(CatFormulaComponentes entity: objs) {  
 DominioCatFormulaComponentes dom = new DominioCatFormulaComponentes();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormulaComponentes.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatFormulaComponentes dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatFormulaComponentes> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatFormulaComponentes> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatFormulaComponentes.findByCatFormulaComponentesId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatFormulaComponentes.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatFormulaComponentes.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatFormulaComponentes entity: objs) {  
 DominioCatFormulaComponentes to = new DominioCatFormulaComponentes();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatFormulaComponentes dom = (DominioCatFormulaComponentes)obj;  
 CatFormulaComponentes entity = new CatFormulaComponentes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFormulaComponentes.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormulaComponentes.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatFormulaComponentes dom = (DominioCatFormulaComponentes)obj;  
 CatFormulaComponentes entity = new CatFormulaComponentes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFormulaComponentes.findById(entity.getCatFormulaComponentesId()).  
 orElse(new CatFormulaComponentes());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFormulaComponentes.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormulaComponentes.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatFormulaComponentes dom = (DominioCatFormulaComponentes)obj;  
 CatFormulaComponentes entity = new CatFormulaComponentes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFormulaComponentes.findById(entity.getCatFormulaComponentesId()).  
 orElse(new CatFormulaComponentes());  
 daoCatFormulaComponentes.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormulaComponentes.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatFormulaComponentes dom = (DominioCatFormulaComponentes)obj;  
 CatFormulaComponentes entity = new CatFormulaComponentes();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFormulaComponentes.findById(entity.getCatFormulaComponentesId()).  
 orElse(new CatFormulaComponentes());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormulaComponentes.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatFormulaComponentes.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormulaComponentes.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatFormulaComponentes existsById(Object obj) throws Exception {  
 try {  
 DominioCatFormulaComponentes dom = (DominioCatFormulaComponentes)obj;  
 CatFormulaComponentes entity = new CatFormulaComponentes();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatFormulaComponentes.existsById(entity.getCatFormulaComponentesId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormulaComponentes.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatFormulaComponentesTxEvento.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatFormulaComponentesTxEvento;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatFormulaComponentesTxEvento;  
import mx.mbi.dgc.jpa.entity.CatFormulaComponentesTxEvento;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatFormulaComponentesTxEvento implements IfaceCatFormulaComponentesTxEvento {  
 @Autowired  
 private DAOCatFormulaComponentesTxEvento daoCatFormulaComponentesTxEvento;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatFormulaComponentesTxEvento> lista = new ArrayList<>();  
 List<CatFormulaComponentesTxEvento> objs = daoCatFormulaComponentesTxEvento.findAll();  
 for(CatFormulaComponentesTxEvento entity: objs) {  
 DominioCatFormulaComponentesTxEvento dom = new DominioCatFormulaComponentesTxEvento();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormulaComponentesTxEvento.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatFormulaComponentesTxEvento dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatFormulaComponentesTxEvento> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatFormulaComponentesTxEvento> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatFormulaComponentesTxEvento.findByCatFormulaComponentesTxEventoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatFormulaComponentesTxEvento.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatFormulaComponentesTxEvento.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatFormulaComponentesTxEvento entity: objs) {  
 DominioCatFormulaComponentesTxEvento to = new DominioCatFormulaComponentesTxEvento();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatFormulaComponentesTxEvento dom = (DominioCatFormulaComponentesTxEvento)obj;  
 CatFormulaComponentesTxEvento entity = new CatFormulaComponentesTxEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFormulaComponentesTxEvento.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormulaComponentesTxEvento.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatFormulaComponentesTxEvento dom = (DominioCatFormulaComponentesTxEvento)obj;  
 CatFormulaComponentesTxEvento entity = new CatFormulaComponentesTxEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFormulaComponentesTxEvento.findById(entity.getCatFormulaComponentesTxEventoId()).  
 orElse(new CatFormulaComponentesTxEvento());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFormulaComponentesTxEvento.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormulaComponentesTxEvento.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatFormulaComponentesTxEvento dom = (DominioCatFormulaComponentesTxEvento)obj;  
 CatFormulaComponentesTxEvento entity = new CatFormulaComponentesTxEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFormulaComponentesTxEvento.findById(entity.getCatFormulaComponentesTxEventoId()).  
 orElse(new CatFormulaComponentesTxEvento());  
 daoCatFormulaComponentesTxEvento.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormulaComponentesTxEvento.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatFormulaComponentesTxEvento dom = (DominioCatFormulaComponentesTxEvento)obj;  
 CatFormulaComponentesTxEvento entity = new CatFormulaComponentesTxEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatFormulaComponentesTxEvento.findById(entity.getCatFormulaComponentesTxEventoId()).  
 orElse(new CatFormulaComponentesTxEvento());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormulaComponentesTxEvento.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatFormulaComponentesTxEvento.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormulaComponentesTxEvento.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatFormulaComponentesTxEvento existsById(Object obj) throws Exception {  
 try {  
 DominioCatFormulaComponentesTxEvento dom = (DominioCatFormulaComponentesTxEvento)obj;  
 CatFormulaComponentesTxEvento entity = new CatFormulaComponentesTxEvento();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatFormulaComponentesTxEvento.existsById(entity.getCatFormulaComponentesTxEventoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatFormulaComponentesTxEvento.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatImpuestos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatImpuestos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatImpuestos;  
import mx.mbi.dgc.jpa.entity.CatImpuestos;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatImpuestos implements IfaceCatImpuestos {  
 @Autowired  
 private DAOCatImpuestos daoCatImpuestos;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatImpuestos> lista = new ArrayList<>();  
 List<CatImpuestos> objs = daoCatImpuestos.findAll();  
 for(CatImpuestos entity: objs) {  
 DominioCatImpuestos dom = new DominioCatImpuestos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatImpuestos.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatImpuestos dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatImpuestos> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatImpuestos> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatImpuestos.findByImpuestoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatImpuestos.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatImpuestos.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatImpuestos entity: objs) {  
 DominioCatImpuestos to = new DominioCatImpuestos();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatImpuestos dom = (DominioCatImpuestos)obj;  
 CatImpuestos entity = new CatImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatImpuestos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatImpuestos.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatImpuestos dom = (DominioCatImpuestos)obj;  
 CatImpuestos entity = new CatImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatImpuestos.findById(entity.getImpuestoId()).  
 orElse(new CatImpuestos());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatImpuestos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatImpuestos.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatImpuestos dom = (DominioCatImpuestos)obj;  
 CatImpuestos entity = new CatImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatImpuestos.findById(entity.getImpuestoId()).  
 orElse(new CatImpuestos());  
 daoCatImpuestos.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatImpuestos.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatImpuestos dom = (DominioCatImpuestos)obj;  
 CatImpuestos entity = new CatImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatImpuestos.findById(entity.getImpuestoId()).  
 orElse(new CatImpuestos());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatImpuestos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatImpuestos.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatImpuestos.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatImpuestos existsById(Object obj) throws Exception {  
 try {  
 DominioCatImpuestos dom = (DominioCatImpuestos)obj;  
 CatImpuestos entity = new CatImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatImpuestos.existsById(entity.getImpuestoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatImpuestos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatLimitesDominio.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatLimitesDominio;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatLimitesDominio;  
import mx.mbi.dgc.jpa.entity.CatLimitesDominio;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatLimitesDominio implements IfaceCatLimitesDominio {  
 @Autowired  
 private DAOCatLimitesDominio daoCatLimitesDominio;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatLimitesDominio> lista = new ArrayList<>();  
 List<CatLimitesDominio> objs = daoCatLimitesDominio.findAll();  
 for(CatLimitesDominio entity: objs) {  
 DominioCatLimitesDominio dom = new DominioCatLimitesDominio();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatLimitesDominio.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatLimitesDominio dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatLimitesDominio> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatLimitesDominio> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatLimitesDominio.findByLimitesDominioId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatLimitesDominio.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatLimitesDominio.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatLimitesDominio entity: objs) {  
 DominioCatLimitesDominio to = new DominioCatLimitesDominio();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatLimitesDominio dom = (DominioCatLimitesDominio)obj;  
 CatLimitesDominio entity = new CatLimitesDominio();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatLimitesDominio.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatLimitesDominio.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatLimitesDominio dom = (DominioCatLimitesDominio)obj;  
 CatLimitesDominio entity = new CatLimitesDominio();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatLimitesDominio.findById(entity.getLimitesDominioId()).  
 orElse(new CatLimitesDominio());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatLimitesDominio.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatLimitesDominio.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatLimitesDominio dom = (DominioCatLimitesDominio)obj;  
 CatLimitesDominio entity = new CatLimitesDominio();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatLimitesDominio.findById(entity.getLimitesDominioId()).  
 orElse(new CatLimitesDominio());  
 daoCatLimitesDominio.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatLimitesDominio.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatLimitesDominio dom = (DominioCatLimitesDominio)obj;  
 CatLimitesDominio entity = new CatLimitesDominio();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatLimitesDominio.findById(entity.getLimitesDominioId()).  
 orElse(new CatLimitesDominio());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatLimitesDominio.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatLimitesDominio.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatLimitesDominio.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatLimitesDominio existsById(Object obj) throws Exception {  
 try {  
 DominioCatLimitesDominio dom = (DominioCatLimitesDominio)obj;  
 CatLimitesDominio entity = new CatLimitesDominio();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatLimitesDominio.existsById(entity.getLimitesDominioId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatLimitesDominio.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatMedidor.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatMedidor;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatMedidor;  
import mx.mbi.dgc.jpa.entity.CatMedidor;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatMedidor implements IfaceCatMedidor {  
 @Autowired  
 private DAOCatMedidor daoCatMedidor;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatMedidor> lista = new ArrayList<>();  
 List<CatMedidor> objs = daoCatMedidor.findAll();  
 for(CatMedidor entity: objs) {  
 DominioCatMedidor dom = new DominioCatMedidor();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMedidor.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatMedidor dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatMedidor> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatMedidor> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatMedidor.findByMedidorId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatMedidor.findByMedidorContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatMedidor.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatMedidor entity: objs) {  
 DominioCatMedidor to = new DominioCatMedidor();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatMedidor dom = (DominioCatMedidor)obj;  
 CatMedidor entity = new CatMedidor();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMedidor.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMedidor.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatMedidor dom = (DominioCatMedidor)obj;  
 CatMedidor entity = new CatMedidor();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMedidor.findById(entity.getMedidorId()).  
 orElse(new CatMedidor());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMedidor.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMedidor.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatMedidor dom = (DominioCatMedidor)obj;  
 CatMedidor entity = new CatMedidor();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMedidor.findById(entity.getMedidorId()).  
 orElse(new CatMedidor());  
 daoCatMedidor.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMedidor.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatMedidor dom = (DominioCatMedidor)obj;  
 CatMedidor entity = new CatMedidor();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMedidor.findById(entity.getMedidorId()).  
 orElse(new CatMedidor());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMedidor.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatMedidor.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMedidor.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatMedidor existsById(Object obj) throws Exception {  
 try {  
 DominioCatMedidor dom = (DominioCatMedidor)obj;  
 CatMedidor entity = new CatMedidor();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatMedidor.existsById(entity.getMedidorId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMedidor.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatMedidores.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatMedidores;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatMedidores;  
import mx.mbi.dgc.jpa.entity.CatMedidores;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatMedidores implements IfaceCatMedidores {  
 @Autowired  
 private DAOCatMedidores daoCatMedidores;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatMedidores> lista = new ArrayList<>();  
 List<CatMedidores> objs = daoCatMedidores.findAll();  
 for(CatMedidores entity: objs) {  
 DominioCatMedidores dom = new DominioCatMedidores();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMedidores.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatMedidores dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatMedidores> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatMedidores> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatMedidores.findByMedidorId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatMedidores.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatMedidores.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatMedidores entity: objs) {  
 DominioCatMedidores to = new DominioCatMedidores();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatMedidores dom = (DominioCatMedidores)obj;  
 CatMedidores entity = new CatMedidores();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMedidores.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMedidores.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatMedidores dom = (DominioCatMedidores)obj;  
 CatMedidores entity = new CatMedidores();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMedidores.findById(entity.getMedidorId()).  
 orElse(new CatMedidores());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMedidores.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMedidores.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatMedidores dom = (DominioCatMedidores)obj;  
 CatMedidores entity = new CatMedidores();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMedidores.findById(entity.getMedidorId()).  
 orElse(new CatMedidores());  
 daoCatMedidores.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMedidores.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatMedidores dom = (DominioCatMedidores)obj;  
 CatMedidores entity = new CatMedidores();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMedidores.findById(entity.getMedidorId()).  
 orElse(new CatMedidores());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMedidores.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatMedidores.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMedidores.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatMedidores existsById(Object obj) throws Exception {  
 try {  
 DominioCatMedidores dom = (DominioCatMedidores)obj;  
 CatMedidores entity = new CatMedidores();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatMedidores.existsById(entity.getMedidorId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMedidores.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatMesCargo.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatMesCargo;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatMesCargo;  
import mx.mbi.dgc.jpa.entity.CatMesCargo;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatMesCargo implements IfaceCatMesCargo {  
 @Autowired  
 private DAOCatMesCargo daoCatMesCargo;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatMesCargo> lista = new ArrayList<>();  
 List<CatMesCargo> objs = daoCatMesCargo.findAll();  
 for(CatMesCargo entity: objs) {  
 DominioCatMesCargo dom = new DominioCatMesCargo();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesCargo.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatMesCargo dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatMesCargo> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatMesCargo> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatMesCargo.findByMesId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatMesCargo.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatMesCargo.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatMesCargo entity: objs) {  
 DominioCatMesCargo to = new DominioCatMesCargo();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatMesCargo dom = (DominioCatMesCargo)obj;  
 CatMesCargo entity = new CatMesCargo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMesCargo.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesCargo.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatMesCargo dom = (DominioCatMesCargo)obj;  
 CatMesCargo entity = new CatMesCargo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMesCargo.findById(entity.getMesId()).  
 orElse(new CatMesCargo());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMesCargo.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesCargo.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatMesCargo dom = (DominioCatMesCargo)obj;  
 CatMesCargo entity = new CatMesCargo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMesCargo.findById(entity.getMesId()).  
 orElse(new CatMesCargo());  
 daoCatMesCargo.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesCargo.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatMesCargo dom = (DominioCatMesCargo)obj;  
 CatMesCargo entity = new CatMesCargo();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMesCargo.findById(entity.getMesId()).  
 orElse(new CatMesCargo());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesCargo.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatMesCargo.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesCargo.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatMesCargo existsById(Object obj) throws Exception {  
 try {  
 DominioCatMesCargo dom = (DominioCatMesCargo)obj;  
 CatMesCargo entity = new CatMesCargo();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatMesCargo.existsById(entity.getMesId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesCargo.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatMesDia.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatMesDia;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatMesDia;  
import mx.mbi.dgc.jpa.entity.CatMesDia;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatMesDia implements IfaceCatMesDia {  
 @Autowired  
 private DAOCatMesDia daoCatMesDia;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatMesDia> lista = new ArrayList<>();  
 List<CatMesDia> objs = daoCatMesDia.findAll();  
 for(CatMesDia entity: objs) {  
 DominioCatMesDia dom = new DominioCatMesDia();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDia.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatMesDia dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatMesDia> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatMesDia> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatMesDia.findByMesDiaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatMesDia.findByDescripcionContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatMesDia.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatMesDia entity: objs) {  
 DominioCatMesDia to = new DominioCatMesDia();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatMesDia dom = (DominioCatMesDia)obj;  
 CatMesDia entity = new CatMesDia();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMesDia.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDia.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatMesDia dom = (DominioCatMesDia)obj;  
 CatMesDia entity = new CatMesDia();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMesDia.findById(entity.getMesDiaId()).  
 orElse(new CatMesDia());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMesDia.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDia.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatMesDia dom = (DominioCatMesDia)obj;  
 CatMesDia entity = new CatMesDia();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMesDia.findById(entity.getMesDiaId()).  
 orElse(new CatMesDia());  
 daoCatMesDia.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDia.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatMesDia dom = (DominioCatMesDia)obj;  
 CatMesDia entity = new CatMesDia();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMesDia.findById(entity.getMesDiaId()).  
 orElse(new CatMesDia());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDia.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatMesDia.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDia.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatMesDia existsById(Object obj) throws Exception {  
 try {  
 DominioCatMesDia dom = (DominioCatMesDia)obj;  
 CatMesDia entity = new CatMesDia();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatMesDia.existsById(entity.getMesDiaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDia.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatMesDiaFirma.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatMesDiaFirma;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatMesDiaFirma;  
import mx.mbi.dgc.jpa.entity.CatMesDiaFirma;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatMesDiaFirma implements IfaceCatMesDiaFirma {  
 @Autowired  
 private DAOCatMesDiaFirma daoCatMesDiaFirma;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatMesDiaFirma> lista = new ArrayList<>();  
 List<CatMesDiaFirma> objs = daoCatMesDiaFirma.findAll();  
 for(CatMesDiaFirma entity: objs) {  
 DominioCatMesDiaFirma dom = new DominioCatMesDiaFirma();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatMesDiaFirma dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatMesDiaFirma> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatMesDiaFirma> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatMesDiaFirma.findByMesDiaFirmaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatMesDiaFirma.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatMesDiaFirma.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatMesDiaFirma entity: objs) {  
 DominioCatMesDiaFirma to = new DominioCatMesDiaFirma();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatMesDiaFirma dom = (DominioCatMesDiaFirma)obj;  
 CatMesDiaFirma entity = new CatMesDiaFirma();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMesDiaFirma.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatMesDiaFirma dom = (DominioCatMesDiaFirma)obj;  
 CatMesDiaFirma entity = new CatMesDiaFirma();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMesDiaFirma.findById(entity.getMesDiaFirmaId()).  
 orElse(new CatMesDiaFirma());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMesDiaFirma.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatMesDiaFirma dom = (DominioCatMesDiaFirma)obj;  
 CatMesDiaFirma entity = new CatMesDiaFirma();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMesDiaFirma.findById(entity.getMesDiaFirmaId()).  
 orElse(new CatMesDiaFirma());  
 daoCatMesDiaFirma.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatMesDiaFirma dom = (DominioCatMesDiaFirma)obj;  
 CatMesDiaFirma entity = new CatMesDiaFirma();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMesDiaFirma.findById(entity.getMesDiaFirmaId()).  
 orElse(new CatMesDiaFirma());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatMesDiaFirma.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatMesDiaFirma existsById(Object obj) throws Exception {  
 try {  
 DominioCatMesDiaFirma dom = (DominioCatMesDiaFirma)obj;  
 CatMesDiaFirma entity = new CatMesDiaFirma();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatMesDiaFirma.existsById(entity.getMesDiaFirmaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatMetodoPago.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatMetodoPago;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatMetodoPago;  
import mx.mbi.dgc.jpa.entity.CatMetodoPago;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatMetodoPago implements IfaceCatMetodoPago {  
 @Autowired  
 private DAOCatMetodoPago daoCatMetodoPago;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatMetodoPago> lista = new ArrayList<>();  
 List<CatMetodoPago> objs = daoCatMetodoPago.findAll();  
 for(CatMetodoPago entity: objs) {  
 DominioCatMetodoPago dom = new DominioCatMetodoPago();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMetodoPago.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatMetodoPago dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatMetodoPago> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatMetodoPago> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatMetodoPago.findByMetodoPagoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatMetodoPago.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatMetodoPago.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatMetodoPago entity: objs) {  
 DominioCatMetodoPago to = new DominioCatMetodoPago();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatMetodoPago dom = (DominioCatMetodoPago)obj;  
 CatMetodoPago entity = new CatMetodoPago();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMetodoPago.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMetodoPago.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatMetodoPago dom = (DominioCatMetodoPago)obj;  
 CatMetodoPago entity = new CatMetodoPago();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMetodoPago.findById(entity.getMetodoPagoId()).  
 orElse(new CatMetodoPago());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMetodoPago.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMetodoPago.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatMetodoPago dom = (DominioCatMetodoPago)obj;  
 CatMetodoPago entity = new CatMetodoPago();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMetodoPago.findById(entity.getMetodoPagoId()).  
 orElse(new CatMetodoPago());  
 daoCatMetodoPago.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMetodoPago.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatMetodoPago dom = (DominioCatMetodoPago)obj;  
 CatMetodoPago entity = new CatMetodoPago();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMetodoPago.findById(entity.getMetodoPagoId()).  
 orElse(new CatMetodoPago());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMetodoPago.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatMetodoPago.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMetodoPago.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatMetodoPago existsById(Object obj) throws Exception {  
 try {  
 DominioCatMetodoPago dom = (DominioCatMetodoPago)obj;  
 CatMetodoPago entity = new CatMetodoPago();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatMetodoPago.existsById(entity.getMetodoPagoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMetodoPago.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatMonedas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatMonedas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatMonedas;  
import mx.mbi.dgc.jpa.entity.CatMonedas;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatMonedas implements IfaceCatMonedas {  
 @Autowired  
 private DAOCatMonedas daoCatMonedas;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatMonedas> lista = new ArrayList<>();  
 List<CatMonedas> objs = daoCatMonedas.findAll();  
 for(CatMonedas entity: objs) {  
 DominioCatMonedas dom = new DominioCatMonedas();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMonedas.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatMonedas dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatMonedas> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatMonedas> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatMonedas.findByMonedaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatMonedas.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatMonedas.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatMonedas entity: objs) {  
 DominioCatMonedas to = new DominioCatMonedas();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatMonedas dom = (DominioCatMonedas)obj;  
 CatMonedas entity = new CatMonedas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMonedas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMonedas.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatMonedas dom = (DominioCatMonedas)obj;  
 CatMonedas entity = new CatMonedas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMonedas.findById(entity.getMonedaId()).  
 orElse(new CatMonedas());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMonedas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMonedas.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatMonedas dom = (DominioCatMonedas)obj;  
 CatMonedas entity = new CatMonedas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMonedas.findById(entity.getMonedaId()).  
 orElse(new CatMonedas());  
 daoCatMonedas.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMonedas.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatMonedas dom = (DominioCatMonedas)obj;  
 CatMonedas entity = new CatMonedas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatMonedas.findById(entity.getMonedaId()).  
 orElse(new CatMonedas());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMonedas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatMonedas.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMonedas.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatMonedas existsById(Object obj) throws Exception {  
 try {  
 DominioCatMonedas dom = (DominioCatMonedas)obj;  
 CatMonedas entity = new CatMonedas();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatMonedas.existsById(entity.getMonedaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMonedas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatOpcionPlanta.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatOpcionPlanta;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatOpcionPlanta;  
import mx.mbi.dgc.jpa.entity.CatOpcionPlanta;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
  
import mx.mbi.dgc.dao.DAOCatDetalleOpciones;  
import mx.mbi.dgc.dao.DAOCatPlanta;  
import mx.mbi.dgc.dominio.DominioCatDetalleOpciones;  
import mx.mbi.dgc.dominio.DominioCatPlanta;  
import mx.mbi.dgc.jpa.entity.CatDetalleOpciones;  
import mx.mbi.dgc.jpa.entity.CatDetalles;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatOpcionPlanta implements IfaceCatOpcionPlanta {  
 @Autowired  
 private DAOCatOpcionPlanta daoCatOpcionPlanta;  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatOpcionPlanta> lista = new ArrayList<>();  
 List<CatOpcionPlanta> objs = daoCatOpcionPlanta.findAll();  
 for(CatOpcionPlanta entity: objs) {  
 DominioCatOpcionPlanta dom = new DominioCatOpcionPlanta();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatOpcionPlanta.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatOpcionPlanta dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatOpcionPlanta> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatOpcionPlanta> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatOpcionPlanta.findByOpcionPlantaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatOpcionPlanta.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatOpcionPlanta.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatOpcionPlanta entity: objs) {  
 DominioCatOpcionPlanta to = new DominioCatOpcionPlanta();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
   
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatOpcionPlanta dom = (DominioCatOpcionPlanta)obj;  
 CatOpcionPlanta entity = new CatOpcionPlanta();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatOpcionPlanta.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return null;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatOpcionPlanta.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatOpcionPlanta dom = (DominioCatOpcionPlanta)obj;  
 CatOpcionPlanta entity = new CatOpcionPlanta();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatOpcionPlanta.findById(entity.getOpcionPlantaId()).  
 orElse(new CatOpcionPlanta());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatOpcionPlanta.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatOpcionPlanta.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatOpcionPlanta dom = (DominioCatOpcionPlanta)obj;  
 CatOpcionPlanta entity = new CatOpcionPlanta();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatOpcionPlanta.findById(entity.getOpcionPlantaId()).  
 orElse(new CatOpcionPlanta());  
 daoCatOpcionPlanta.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return obj;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatOpcionPlanta.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatOpcionPlanta dom = (DominioCatOpcionPlanta)obj;  
 CatOpcionPlanta entity = new CatOpcionPlanta();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatOpcionPlanta.findById(entity.getOpcionPlantaId()).  
 orElse(new CatOpcionPlanta());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatOpcionPlanta.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatOpcionPlanta.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatOpcionPlanta.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatOpcionPlanta existsById(Object obj) throws Exception {  
 try {  
 DominioCatOpcionPlanta dom = (DominioCatOpcionPlanta)obj;  
 CatOpcionPlanta entity = new CatOpcionPlanta();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatOpcionPlanta.existsById(entity.getOpcionPlantaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatOpcionPlanta.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
   
 @Override  
 public DominioCatOpcionPlanta findByOpcionPlanta(DominioCatOpcionPlanta dominioCatOpcionPlanta) throws Exception {  
 CatPlanta catPlanta = new CatPlanta();  
 CatDetalleOpciones catDetalleOpciones = new CatDetalleOpciones();  
 JRUtil.copyProperties(catPlanta, dominioCatOpcionPlanta.getPlantaId());  
 JRUtil.copyProperties(catDetalleOpciones, dominioCatOpcionPlanta.getDetalleOpcionesId());  
 List<CatOpcionPlanta> listaPlanta = daoCatOpcionPlanta.findByPlantaId(catPlanta);  
 List<CatOpcionPlanta> listaOpcion = daoCatOpcionPlanta.findByDetalleOpcionesId(catDetalleOpciones);  
 for (CatOpcionPlanta aux : listaPlanta)  
 for(CatOpcionPlanta aux2 : listaOpcion){  
 if(aux.getOpcionPlantaId() == aux2.getOpcionPlantaId()){  
 DominioCatOpcionPlanta lista2 = new DominioCatOpcionPlanta();  
 JRUtil.copyProperties(lista2, aux);  
 dominioCatOpcionPlanta = lista2 ;  
 System.out.println("planta" + aux2.getDetalleOpcionesId().getDetalleOpcionesId() + "detalle" +aux.getDetalleOpcionesId());  
 }  
 }  
 return dominioCatOpcionPlanta;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatPaises.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatPaises;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatPaises;  
import mx.mbi.dgc.jpa.entity.CatPaises;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatPaises implements IfaceCatPaises {  
 @Autowired  
 private DAOCatPaises daoCatPaises;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatPaises> lista = new ArrayList<>();  
 List<CatPaises> objs = daoCatPaises.findAll();  
 for(CatPaises entity: objs) {  
 DominioCatPaises dom = new DominioCatPaises();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPaises.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatPaises dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatPaises> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatPaises> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatPaises.findByPaisId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatPaises.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatPaises.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatPaises entity: objs) {  
 DominioCatPaises to = new DominioCatPaises();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatPaises dom = (DominioCatPaises)obj;  
 CatPaises entity = new CatPaises();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPaises.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPaises.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatPaises dom = (DominioCatPaises)obj;  
 CatPaises entity = new CatPaises();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPaises.findById(entity.getPaisId()).  
 orElse(new CatPaises());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPaises.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPaises.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatPaises dom = (DominioCatPaises)obj;  
 CatPaises entity = new CatPaises();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPaises.findById(entity.getPaisId()).  
 orElse(new CatPaises());  
 daoCatPaises.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPaises.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatPaises dom = (DominioCatPaises)obj;  
 CatPaises entity = new CatPaises();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPaises.findById(entity.getPaisId()).  
 orElse(new CatPaises());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPaises.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatPaises.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPaises.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatPaises existsById(Object obj) throws Exception {  
 try {  
 DominioCatPaises dom = (DominioCatPaises)obj;  
 CatPaises entity = new CatPaises();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatPaises.existsById(entity.getPaisId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPaises.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatPermiso.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatPermiso;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatPermiso;  
import mx.mbi.dgc.jpa.entity.CatPermiso;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatPermiso implements IfaceCatPermiso {  
 @Autowired  
 private DAOCatPermiso daoCatPermiso;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatPermiso> lista = new ArrayList<>();  
 List<CatPermiso> objs = daoCatPermiso.findAll();  
 for(CatPermiso entity: objs) {  
 DominioCatPermiso dom = new DominioCatPermiso();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPermiso.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatPermiso dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatPermiso> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatPermiso> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatPermiso.findByPermisoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatPermiso.findByPermisoContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatPermiso.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatPermiso entity: objs) {  
 DominioCatPermiso to = new DominioCatPermiso();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatPermiso dom = (DominioCatPermiso)obj;  
 CatPermiso entity = new CatPermiso();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPermiso.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPermiso.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatPermiso dom = (DominioCatPermiso)obj;  
 CatPermiso entity = new CatPermiso();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPermiso.findById(entity.getPermisoId()).  
 orElse(new CatPermiso());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPermiso.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPermiso.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatPermiso dom = (DominioCatPermiso)obj;  
 CatPermiso entity = new CatPermiso();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPermiso.findById(entity.getPermisoId()).  
 orElse(new CatPermiso());  
 daoCatPermiso.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPermiso.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatPermiso dom = (DominioCatPermiso)obj;  
 CatPermiso entity = new CatPermiso();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPermiso.findById(entity.getPermisoId()).  
 orElse(new CatPermiso());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPermiso.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatPermiso.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPermiso.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatPermiso existsById(Object obj) throws Exception {  
 try {  
 DominioCatPermiso dom = (DominioCatPermiso)obj;  
 CatPermiso entity = new CatPermiso();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatPermiso.existsById(entity.getPermisoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPermiso.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatPlanta.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatPlanta;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatPlanta;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatPlanta implements IfaceCatPlanta {  
 @Autowired  
 private DAOCatPlanta daoCatPlanta;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatPlanta> lista = new ArrayList<>();  
 List<CatPlanta> objs = daoCatPlanta.findAll();  
 for(CatPlanta entity: objs) {  
 DominioCatPlanta dom = new DominioCatPlanta();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPlanta.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatPlanta dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatPlanta> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatPlanta> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatPlanta.findByPlantaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatPlanta.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatPlanta.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatPlanta entity: objs) {  
 DominioCatPlanta to = new DominioCatPlanta();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatPlanta dom = (DominioCatPlanta)obj;  
 CatPlanta entity = new CatPlanta();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPlanta.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPlanta.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatPlanta dom = (DominioCatPlanta)obj;  
 CatPlanta entity = new CatPlanta();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPlanta.findById(entity.getPlantaId()).  
 orElse(new CatPlanta());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPlanta.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPlanta.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatPlanta dom = (DominioCatPlanta)obj;  
 CatPlanta entity = new CatPlanta();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPlanta.findById(entity.getPlantaId()).  
 orElse(new CatPlanta());  
 daoCatPlanta.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPlanta.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatPlanta dom = (DominioCatPlanta)obj;  
 CatPlanta entity = new CatPlanta();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPlanta.findById(entity.getPlantaId()).  
 orElse(new CatPlanta());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPlanta.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatPlanta.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPlanta.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatPlanta existsById(Object obj) throws Exception {  
 try {  
 DominioCatPlanta dom = (DominioCatPlanta)obj;  
 CatPlanta entity = new CatPlanta();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatPlanta.existsById(entity.getPlantaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPlanta.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatPlantaTipoFacturas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatPlantaTipoFacturas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatPlantaTipoFacturas;  
import mx.mbi.dgc.jpa.entity.CatPlantaTipoFacturas;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatPlantaTipoFacturas implements IfaceCatPlantaTipoFacturas {  
 @Autowired  
 private DAOCatPlantaTipoFacturas daoCatPlantaTipoFacturas;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatPlantaTipoFacturas> lista = new ArrayList<>();  
 List<CatPlantaTipoFacturas> objs = daoCatPlantaTipoFacturas.findAll();  
 for(CatPlantaTipoFacturas entity: objs) {  
 DominioCatPlantaTipoFacturas dom = new DominioCatPlantaTipoFacturas();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPlantaTipoFacturas.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatPlantaTipoFacturas dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatPlantaTipoFacturas> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatPlantaTipoFacturas> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatPlantaTipoFacturas.findByPlantaTipoFacturaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatPlantaTipoFacturas.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatPlantaTipoFacturas.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatPlantaTipoFacturas entity: objs) {  
 DominioCatPlantaTipoFacturas to = new DominioCatPlantaTipoFacturas();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatPlantaTipoFacturas dom = (DominioCatPlantaTipoFacturas)obj;  
 CatPlantaTipoFacturas entity = new CatPlantaTipoFacturas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPlantaTipoFacturas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPlantaTipoFacturas.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatPlantaTipoFacturas dom = (DominioCatPlantaTipoFacturas)obj;  
 CatPlantaTipoFacturas entity = new CatPlantaTipoFacturas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPlantaTipoFacturas.findById(entity.getPlantaTipoFacturaId()).  
 orElse(new CatPlantaTipoFacturas());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPlantaTipoFacturas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPlantaTipoFacturas.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatPlantaTipoFacturas dom = (DominioCatPlantaTipoFacturas)obj;  
 CatPlantaTipoFacturas entity = new CatPlantaTipoFacturas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPlantaTipoFacturas.findById(entity.getPlantaTipoFacturaId()).  
 orElse(new CatPlantaTipoFacturas());  
 daoCatPlantaTipoFacturas.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPlantaTipoFacturas.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatPlantaTipoFacturas dom = (DominioCatPlantaTipoFacturas)obj;  
 CatPlantaTipoFacturas entity = new CatPlantaTipoFacturas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPlantaTipoFacturas.findById(entity.getPlantaTipoFacturaId()).  
 orElse(new CatPlantaTipoFacturas());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPlantaTipoFacturas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatPlantaTipoFacturas.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPlantaTipoFacturas.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatPlantaTipoFacturas existsById(Object obj) throws Exception {  
 try {  
 DominioCatPlantaTipoFacturas dom = (DominioCatPlantaTipoFacturas)obj;  
 CatPlantaTipoFacturas entity = new CatPlantaTipoFacturas();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatPlantaTipoFacturas.existsById(entity.getPlantaTipoFacturaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPlantaTipoFacturas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatPropCincominutal.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatPropCincominutal;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatPropCincominutal;  
import mx.mbi.dgc.jpa.entity.CatPropCincominutal;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatPropCincominutal implements IfaceCatPropCincominutal {  
 @Autowired  
 private DAOCatPropCincominutal daoCatPropCincominutal;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatPropCincominutal> lista = new ArrayList<>();  
 List<CatPropCincominutal> objs = daoCatPropCincominutal.findAll();  
 for(CatPropCincominutal entity: objs) {  
 DominioCatPropCincominutal dom = new DominioCatPropCincominutal();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPropCincominutal.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatPropCincominutal dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatPropCincominutal> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatPropCincominutal> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatPropCincominutal.findByPropiedadId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatPropCincominutal.findByPropiedadContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatPropCincominutal.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatPropCincominutal entity: objs) {  
 DominioCatPropCincominutal to = new DominioCatPropCincominutal();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatPropCincominutal dom = (DominioCatPropCincominutal)obj;  
 CatPropCincominutal entity = new CatPropCincominutal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPropCincominutal.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPropCincominutal.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatPropCincominutal dom = (DominioCatPropCincominutal)obj;  
 CatPropCincominutal entity = new CatPropCincominutal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPropCincominutal.findById(entity.getPropiedadId()).  
 orElse(new CatPropCincominutal());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPropCincominutal.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPropCincominutal.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatPropCincominutal dom = (DominioCatPropCincominutal)obj;  
 CatPropCincominutal entity = new CatPropCincominutal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPropCincominutal.findById(entity.getPropiedadId()).  
 orElse(new CatPropCincominutal());  
 daoCatPropCincominutal.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPropCincominutal.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatPropCincominutal dom = (DominioCatPropCincominutal)obj;  
 CatPropCincominutal entity = new CatPropCincominutal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatPropCincominutal.findById(entity.getPropiedadId()).  
 orElse(new CatPropCincominutal());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPropCincominutal.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatPropCincominutal.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPropCincominutal.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatPropCincominutal existsById(Object obj) throws Exception {  
 try {  
 DominioCatPropCincominutal dom = (DominioCatPropCincominutal)obj;  
 CatPropCincominutal entity = new CatPropCincominutal();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatPropCincominutal.existsById(entity.getPropiedadId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatPropCincominutal.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatQuery.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatQuery;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatQuery;  
import mx.mbi.dgc.jpa.entity.CatQuery;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatQuery implements IfaceCatQuery {  
 @Autowired  
 private DAOCatQuery daoCatQuery;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatQuery> lista = new ArrayList<>();  
 List<CatQuery> objs = daoCatQuery.findAll();  
 for(CatQuery entity: objs) {  
 DominioCatQuery dom = new DominioCatQuery();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatQuery.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatQuery dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatQuery> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatQuery> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatQuery.findByQueryId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatQuery.findByQueryContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatQuery.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatQuery entity: objs) {  
 DominioCatQuery to = new DominioCatQuery();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatQuery dom = (DominioCatQuery)obj;  
 CatQuery entity = new CatQuery();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatQuery.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatQuery.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatQuery dom = (DominioCatQuery)obj;  
 CatQuery entity = new CatQuery();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatQuery.findById(entity.getQueryId()).  
 orElse(new CatQuery());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatQuery.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatQuery.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatQuery dom = (DominioCatQuery)obj;  
 CatQuery entity = new CatQuery();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatQuery.findById(entity.getQueryId()).  
 orElse(new CatQuery());  
 daoCatQuery.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatQuery.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatQuery dom = (DominioCatQuery)obj;  
 CatQuery entity = new CatQuery();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatQuery.findById(entity.getQueryId()).  
 orElse(new CatQuery());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatQuery.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatQuery.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatQuery.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatQuery existsById(Object obj) throws Exception {  
 try {  
 DominioCatQuery dom = (DominioCatQuery)obj;  
 CatQuery entity = new CatQuery();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatQuery.existsById(entity.getQueryId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatQuery.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatRangoVariable.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatRangoVariable;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatRangoVariable;  
import mx.mbi.dgc.jpa.entity.CatRangoVariable;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatRangoVariable implements IfaceCatRangoVariable {  
 @Autowired  
 private DAOCatRangoVariable daoCatRangoVariable;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatRangoVariable> lista = new ArrayList<>();  
 List<CatRangoVariable> objs = daoCatRangoVariable.findAll();  
 for(CatRangoVariable entity: objs) {  
 DominioCatRangoVariable dom = new DominioCatRangoVariable();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRangoVariable.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatRangoVariable dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatRangoVariable> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatRangoVariable> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatRangoVariable.findByRangoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatRangoVariable.findByUnificacionEscalaContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatRangoVariable.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatRangoVariable entity: objs) {  
 DominioCatRangoVariable to = new DominioCatRangoVariable();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatRangoVariable dom = (DominioCatRangoVariable)obj;  
 CatRangoVariable entity = new CatRangoVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatRangoVariable.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRangoVariable.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatRangoVariable dom = (DominioCatRangoVariable)obj;  
 CatRangoVariable entity = new CatRangoVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatRangoVariable.findById(entity.getRangoId()).  
 orElse(new CatRangoVariable());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatRangoVariable.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRangoVariable.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatRangoVariable dom = (DominioCatRangoVariable)obj;  
 CatRangoVariable entity = new CatRangoVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatRangoVariable.findById(entity.getRangoId()).  
 orElse(new CatRangoVariable());  
 daoCatRangoVariable.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRangoVariable.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatRangoVariable dom = (DominioCatRangoVariable)obj;  
 CatRangoVariable entity = new CatRangoVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatRangoVariable.findById(entity.getRangoId()).  
 orElse(new CatRangoVariable());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRangoVariable.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatRangoVariable.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRangoVariable.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatRangoVariable existsById(Object obj) throws Exception {  
 try {  
 DominioCatRangoVariable dom = (DominioCatRangoVariable)obj;  
 CatRangoVariable entity = new CatRangoVariable();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatRangoVariable.existsById(entity.getRangoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRangoVariable.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatRegimenFiscal.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatRegimenFiscal;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatRegimenFiscal;  
import mx.mbi.dgc.jpa.entity.CatRegimenFiscal;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatRegimenFiscal implements IfaceCatRegimenFiscal {  
 @Autowired  
 private DAOCatRegimenFiscal daoCatRegimenFiscal;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatRegimenFiscal> lista = new ArrayList<>();  
 List<CatRegimenFiscal> objs = daoCatRegimenFiscal.findAll();  
 for(CatRegimenFiscal entity: objs) {  
 DominioCatRegimenFiscal dom = new DominioCatRegimenFiscal();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRegimenFiscal.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatRegimenFiscal dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatRegimenFiscal> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatRegimenFiscal> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatRegimenFiscal.findByRegimenFiscalId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatRegimenFiscal.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatRegimenFiscal.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatRegimenFiscal entity: objs) {  
 DominioCatRegimenFiscal to = new DominioCatRegimenFiscal();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatRegimenFiscal dom = (DominioCatRegimenFiscal)obj;  
 CatRegimenFiscal entity = new CatRegimenFiscal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatRegimenFiscal.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRegimenFiscal.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatRegimenFiscal dom = (DominioCatRegimenFiscal)obj;  
 CatRegimenFiscal entity = new CatRegimenFiscal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatRegimenFiscal.findById(entity.getRegimenFiscalId()).  
 orElse(new CatRegimenFiscal());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatRegimenFiscal.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRegimenFiscal.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatRegimenFiscal dom = (DominioCatRegimenFiscal)obj;  
 CatRegimenFiscal entity = new CatRegimenFiscal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatRegimenFiscal.findById(entity.getRegimenFiscalId()).  
 orElse(new CatRegimenFiscal());  
 daoCatRegimenFiscal.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRegimenFiscal.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatRegimenFiscal dom = (DominioCatRegimenFiscal)obj;  
 CatRegimenFiscal entity = new CatRegimenFiscal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatRegimenFiscal.findById(entity.getRegimenFiscalId()).  
 orElse(new CatRegimenFiscal());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRegimenFiscal.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatRegimenFiscal.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRegimenFiscal.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatRegimenFiscal existsById(Object obj) throws Exception {  
 try {  
 DominioCatRegimenFiscal dom = (DominioCatRegimenFiscal)obj;  
 CatRegimenFiscal entity = new CatRegimenFiscal();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatRegimenFiscal.existsById(entity.getRegimenFiscalId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRegimenFiscal.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatReglaNegocio.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatReglaNegocio;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatReglaNegocio;  
import mx.mbi.dgc.jpa.entity.CatReglaNegocio;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatReglaNegocio implements IfaceCatReglaNegocio {  
 @Autowired  
 private DAOCatReglaNegocio daoCatReglaNegocio;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatReglaNegocio> lista = new ArrayList<>();  
 List<CatReglaNegocio> objs = daoCatReglaNegocio.findAll();  
 for(CatReglaNegocio entity: objs) {  
 DominioCatReglaNegocio dom = new DominioCatReglaNegocio();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatReglaNegocio.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatReglaNegocio dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatReglaNegocio> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatReglaNegocio> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatReglaNegocio.findByReglaNegocioId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatReglaNegocio.findByAlgoritmoContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatReglaNegocio.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatReglaNegocio entity: objs) {  
 DominioCatReglaNegocio to = new DominioCatReglaNegocio();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatReglaNegocio dom = (DominioCatReglaNegocio)obj;  
 CatReglaNegocio entity = new CatReglaNegocio();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatReglaNegocio.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatReglaNegocio.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatReglaNegocio dom = (DominioCatReglaNegocio)obj;  
 CatReglaNegocio entity = new CatReglaNegocio();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatReglaNegocio.findById(entity.getReglaNegocioId()).  
 orElse(new CatReglaNegocio());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatReglaNegocio.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatReglaNegocio.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatReglaNegocio dom = (DominioCatReglaNegocio)obj;  
 CatReglaNegocio entity = new CatReglaNegocio();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatReglaNegocio.findById(entity.getReglaNegocioId()).  
 orElse(new CatReglaNegocio());  
 daoCatReglaNegocio.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatReglaNegocio.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatReglaNegocio dom = (DominioCatReglaNegocio)obj;  
 CatReglaNegocio entity = new CatReglaNegocio();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatReglaNegocio.findById(entity.getReglaNegocioId()).  
 orElse(new CatReglaNegocio());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatReglaNegocio.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatReglaNegocio.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatReglaNegocio.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatReglaNegocio existsById(Object obj) throws Exception {  
 try {  
 DominioCatReglaNegocio dom = (DominioCatReglaNegocio)obj;  
 CatReglaNegocio entity = new CatReglaNegocio();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatReglaNegocio.existsById(entity.getReglaNegocioId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatReglaNegocio.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatRol.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatRol;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatRol;  
import mx.mbi.dgc.jpa.entity.CatRol;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatRol implements IfaceCatRol {  
 @Autowired  
 private DAOCatRol daoCatRol;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatRol> lista = new ArrayList<>();  
 List<CatRol> objs = daoCatRol.findAll();  
 for(CatRol entity: objs) {  
 DominioCatRol dom = new DominioCatRol();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRol.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatRol dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatRol> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatRol> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatRol.findByRolId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatRol.findByRolContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatRol.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatRol entity: objs) {  
 DominioCatRol to = new DominioCatRol();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatRol dom = (DominioCatRol)obj;  
 CatRol entity = new CatRol();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatRol.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRol.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatRol dom = (DominioCatRol)obj;  
 CatRol entity = new CatRol();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatRol.findById(entity.getRolId()).  
 orElse(new CatRol());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatRol.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRol.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatRol dom = (DominioCatRol)obj;  
 CatRol entity = new CatRol();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatRol.findById(entity.getRolId()).  
 orElse(new CatRol());  
 daoCatRol.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRol.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatRol dom = (DominioCatRol)obj;  
 CatRol entity = new CatRol();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatRol.findById(entity.getRolId()).  
 orElse(new CatRol());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRol.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatRol.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRol.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatRol existsById(Object obj) throws Exception {  
 try {  
 DominioCatRol dom = (DominioCatRol)obj;  
 CatRol entity = new CatRol();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatRol.existsById(entity.getRolId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatRol.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatServicioImpuestos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatServicioImpuestos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatServicioImpuestos;  
import mx.mbi.dgc.jpa.entity.CatServicioImpuestos;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatServicioImpuestos implements IfaceCatServicioImpuestos {  
 @Autowired  
 private DAOCatServicioImpuestos daoCatServicioImpuestos;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatServicioImpuestos> lista = new ArrayList<>();  
 List<CatServicioImpuestos> objs = daoCatServicioImpuestos.findAll();  
 for(CatServicioImpuestos entity: objs) {  
 DominioCatServicioImpuestos dom = new DominioCatServicioImpuestos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatServicioImpuestos.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatServicioImpuestos dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatServicioImpuestos> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatServicioImpuestos> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatServicioImpuestos.findByServicioImpuestoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatServicioImpuestos.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatServicioImpuestos.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatServicioImpuestos entity: objs) {  
 DominioCatServicioImpuestos to = new DominioCatServicioImpuestos();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatServicioImpuestos dom = (DominioCatServicioImpuestos)obj;  
 CatServicioImpuestos entity = new CatServicioImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatServicioImpuestos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatServicioImpuestos.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatServicioImpuestos dom = (DominioCatServicioImpuestos)obj;  
 CatServicioImpuestos entity = new CatServicioImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatServicioImpuestos.findById(entity.getServicioImpuestoId()).  
 orElse(new CatServicioImpuestos());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatServicioImpuestos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatServicioImpuestos.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatServicioImpuestos dom = (DominioCatServicioImpuestos)obj;  
 CatServicioImpuestos entity = new CatServicioImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatServicioImpuestos.findById(entity.getServicioImpuestoId()).  
 orElse(new CatServicioImpuestos());  
 daoCatServicioImpuestos.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatServicioImpuestos.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatServicioImpuestos dom = (DominioCatServicioImpuestos)obj;  
 CatServicioImpuestos entity = new CatServicioImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatServicioImpuestos.findById(entity.getServicioImpuestoId()).  
 orElse(new CatServicioImpuestos());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatServicioImpuestos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatServicioImpuestos.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatServicioImpuestos.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatServicioImpuestos existsById(Object obj) throws Exception {  
 try {  
 DominioCatServicioImpuestos dom = (DominioCatServicioImpuestos)obj;  
 CatServicioImpuestos entity = new CatServicioImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatServicioImpuestos.existsById(entity.getServicioImpuestoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatServicioImpuestos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatServicios.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatServicios;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatServicios;  
import mx.mbi.dgc.jpa.entity.CatServicios;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatServicios implements IfaceCatServicios {  
 @Autowired  
 private DAOCatServicios daoCatServicios;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatServicios> lista = new ArrayList<>();  
 List<CatServicios> objs = daoCatServicios.findAll();  
 for(CatServicios entity: objs) {  
 DominioCatServicios dom = new DominioCatServicios();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatServicios.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatServicios dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatServicios> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatServicios> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatServicios.findByServicioId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatServicios.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatServicios.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatServicios entity: objs) {  
 DominioCatServicios to = new DominioCatServicios();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatServicios dom = (DominioCatServicios)obj;  
 CatServicios entity = new CatServicios();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatServicios.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatServicios.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatServicios dom = (DominioCatServicios)obj;  
 CatServicios entity = new CatServicios();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatServicios.findById(entity.getServicioId()).  
 orElse(new CatServicios());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatServicios.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatServicios.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatServicios dom = (DominioCatServicios)obj;  
 CatServicios entity = new CatServicios();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatServicios.findById(entity.getServicioId()).  
 orElse(new CatServicios());  
 daoCatServicios.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatServicios.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatServicios dom = (DominioCatServicios)obj;  
 CatServicios entity = new CatServicios();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatServicios.findById(entity.getServicioId()).  
 orElse(new CatServicios());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatServicios.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatServicios.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatServicios.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatServicios existsById(Object obj) throws Exception {  
 try {  
 DominioCatServicios dom = (DominioCatServicios)obj;  
 CatServicios entity = new CatServicios();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatServicios.existsById(entity.getServicioId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatServicios.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatSistemas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatSistemas ;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatSistemas ;  
import mx.mbi.dgc.jpa.entity.CatSistemas ;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatSistemas implements IfaceCatSistemas {  
 @Autowired  
 private DAOCatSistemas daoCatSistemas ;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatSistemas > lista = new ArrayList<>();  
 List<CatSistemas > objs = daoCatSistemas .findAll();  
 for(CatSistemas entity: objs) {  
 DominioCatSistemas dom = new DominioCatSistemas ();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatSistemas .findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatSistemas dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatSistemas > lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatSistemas > objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatSistemas .findBySistemaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatSistemas .findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatSistemas .findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatSistemas entity: objs) {  
 DominioCatSistemas to = new DominioCatSistemas ();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatSistemas dom = (DominioCatSistemas )obj;  
 CatSistemas entity = new CatSistemas ();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatSistemas .save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatSistemas .save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatSistemas dom = (DominioCatSistemas )obj;  
 CatSistemas entity = new CatSistemas ();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatSistemas .findById(entity.getSistemaId()).  
 orElse(new CatSistemas ());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatSistemas .save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatSistemas .update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatSistemas dom = (DominioCatSistemas )obj;  
 CatSistemas entity = new CatSistemas ();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatSistemas .findById(entity.getSistemaId()).  
 orElse(new CatSistemas ());  
 daoCatSistemas .delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatSistemas .delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatSistemas dom = (DominioCatSistemas )obj;  
 CatSistemas entity = new CatSistemas ();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatSistemas .findById(entity.getSistemaId()).  
 orElse(new CatSistemas ());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatSistemas .findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatSistemas .count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatSistemas .count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatSistemas existsById(Object obj) throws Exception {  
 try {  
 DominioCatSistemas dom = (DominioCatSistemas )obj;  
 CatSistemas entity = new CatSistemas ();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatSistemas .existsById(entity.getSistemaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatSistemas .findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatSubCargos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatSubCargos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatSubCargos;  
import mx.mbi.dgc.jpa.entity.CatSubCargos;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatSubCargos implements IfaceCatSubCargos {  
 @Autowired  
 private DAOCatSubCargos daoCatSubCargos;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatSubCargos> lista = new ArrayList<>();  
 List<CatSubCargos> objs = daoCatSubCargos.findAll();  
 for(CatSubCargos entity: objs) {  
 DominioCatSubCargos dom = new DominioCatSubCargos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatSubCargos.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatSubCargos dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatSubCargos> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatSubCargos> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatSubCargos.findBySubCargoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatSubCargos.findBySubCargoContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatSubCargos.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatSubCargos entity: objs) {  
 DominioCatSubCargos to = new DominioCatSubCargos();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatSubCargos dom = (DominioCatSubCargos)obj;  
 CatSubCargos entity = new CatSubCargos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatSubCargos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatSubCargos.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatSubCargos dom = (DominioCatSubCargos)obj;  
 CatSubCargos entity = new CatSubCargos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatSubCargos.findById(entity.getSubCargoId()).  
 orElse(new CatSubCargos());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatSubCargos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatSubCargos.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatSubCargos dom = (DominioCatSubCargos)obj;  
 CatSubCargos entity = new CatSubCargos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatSubCargos.findById(entity.getSubCargoId()).  
 orElse(new CatSubCargos());  
 daoCatSubCargos.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatSubCargos.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatSubCargos dom = (DominioCatSubCargos)obj;  
 CatSubCargos entity = new CatSubCargos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatSubCargos.findById(entity.getSubCargoId()).  
 orElse(new CatSubCargos());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatSubCargos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatSubCargos.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatSubCargos.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatSubCargos existsById(Object obj) throws Exception {  
 try {  
 DominioCatSubCargos dom = (DominioCatSubCargos)obj;  
 CatSubCargos entity = new CatSubCargos();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatSubCargos.existsById(entity.getSubCargoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatSubCargos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatTags.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatTags;  
import mx.mbi.dgc.dominio.DominioCatPlanta;  
import mx.mbi.dgc.dominio.DominioCatTags;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.jpa.entity.CatTags;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCatTags implements IfaceCatTags {  
 @Autowired  
 private DAOCatTags daoCatTags;  
  
 @Override  
 public List<DominioCatTags> findByPlantaId(DominioCatPlanta dom) throws Exception {  
 try {  
 List<DominioCatTags> lista = new ArrayList<>();  
 List<CatTags> objs = daoCatTags.findByPlantaIdOrderByOrdenAsc(dom.getPlantaId());  
 for (CatTags entity : objs) {  
 DominioCatTags dom2 = new DominioCatTags();  
 JRUtil.copyProperties(dom2, entity);  
 lista.add(dom2);  
 }  
 return lista;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 throw new Exception("Error in BusinessCatTags.findById: " + ex);  
 }  
 }  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatTags> lista = new ArrayList<>();  
 List<CatTags> objs = daoCatTags.findAll();  
 for (CatTags entity : objs) {  
 DominioCatTags dom = new DominioCatTags();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatTags.findAll: " + ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatTags dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatTags> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatTags> objs = null;  
 if (dom.getSearch() != null && dom.getSearch().getValue() != null &&  
 ("" + dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs =  
 daoCatTags.findByTagId(new java.math.BigDecimal("" + dom.getSearch().getValue()),  
 PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC : Sort.Direction.DESC,  
 dom.getCampos()  
 .get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex) {  
 try {  
 objs =  
 daoCatTags.findByTagContaining("" + dom.getSearch().getValue(),  
 PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC : Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex2) {  
 }  
 }  
 } else {  
 objs =  
 daoCatTags.findAll(PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ? Sort.Direction.ASC :  
 Sort.Direction.DESC, dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 }  
 for (CatTags entity : objs) {  
 DominioCatTags to = new DominioCatTags();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: " + ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatTags dom = (DominioCatTags) obj;  
 CatTags entity = new CatTags();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTags.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatTags.save: " + ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatTags dom = (DominioCatTags) obj;  
 CatTags entity = new CatTags();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTags.findById(entity.getTagId()).orElse(new CatTags());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTags.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatTags.update: " + ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatTags dom = (DominioCatTags) obj;  
 CatTags entity = new CatTags();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTags.findById(entity.getTagId()).orElse(new CatTags());  
 daoCatTags.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatTags.delete: " + ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatTags dom = (DominioCatTags) obj;  
 CatTags entity = new CatTags();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTags.findById(entity.getTagId()).orElse(new CatTags());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatTags.findById: " + ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatTags.count();  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatTags.count: " + ex);  
 }  
 }  
  
 @Override  
 public DominioCatTags existsById(Object obj) throws Exception {  
 try {  
 DominioCatTags dom = (DominioCatTags) obj;  
 CatTags entity = new CatTags();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatTags.existsById(entity.getTagId()));  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatTags.findById: " + ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatTipoDato.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatTipoDato;  
import mx.mbi.dgc.dominio.DominioCatTipoDato;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.jpa.entity.CatTipoDato;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCatTipoDato implements IfaceCatTipoDato {  
 @Autowired  
 private DAOCatTipoDato daoCatTipoDato;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatTipoDato> lista = new ArrayList<>();  
 List<CatTipoDato> objs = daoCatTipoDato.findAll();  
 for(CatTipoDato entity: objs) {  
 DominioCatTipoDato dom = new DominioCatTipoDato();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoDato.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatTipoDato dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatTipoDato> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatTipoDato> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatTipoDato.findByTipoDatoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 ex.printStackTrace();  
 try {  
 objs = daoCatTipoDato.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatTipoDato.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatTipoDato entity: objs) {  
 DominioCatTipoDato to = new DominioCatTipoDato();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 ex.printStackTrace();  
 throw new Exception("Error in businessCatTipoDato.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatTipoDato dom = (DominioCatTipoDato)obj;  
 CatTipoDato entity = new CatTipoDato();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoDato.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoDato.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatTipoDato dom = (DominioCatTipoDato)obj;  
 CatTipoDato entity = new CatTipoDato();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoDato.findById(entity.getTipoDatoId()).  
 orElse(new CatTipoDato());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoDato.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoDato.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatTipoDato dom = (DominioCatTipoDato)obj;  
 CatTipoDato entity = new CatTipoDato();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoDato.findById(entity.getTipoDatoId()).  
 orElse(new CatTipoDato());  
 daoCatTipoDato.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoDato.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatTipoDato dom = (DominioCatTipoDato)obj;  
 CatTipoDato entity = new CatTipoDato();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoDato.findById(entity.getTipoDatoId()).  
 orElse(new CatTipoDato());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoDato.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatTipoDato.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoDato.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatTipoDato existsById(Object obj) throws Exception {  
 try {  
 DominioCatTipoDato dom = (DominioCatTipoDato)obj;  
 CatTipoDato entity = new CatTipoDato();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatTipoDato.existsById(entity.getTipoDatoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoDato.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatTipoDocumento.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatTipoDocumento;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatTipoDocumento;  
import mx.mbi.dgc.jpa.entity.CatTipoDocumento;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatTipoDocumento implements IfaceCatTipoDocumento {  
 @Autowired  
 private DAOCatTipoDocumento daoCatTipoDocumento;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatTipoDocumento> lista = new ArrayList<>();  
 List<CatTipoDocumento> objs = daoCatTipoDocumento.findAll();  
 for(CatTipoDocumento entity: objs) {  
 DominioCatTipoDocumento dom = new DominioCatTipoDocumento();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoDocumento.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatTipoDocumento dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatTipoDocumento> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatTipoDocumento> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatTipoDocumento.findByTipoDocumentoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatTipoDocumento.findByTipoContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatTipoDocumento.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatTipoDocumento entity: objs) {  
 DominioCatTipoDocumento to = new DominioCatTipoDocumento();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatTipoDocumento dom = (DominioCatTipoDocumento)obj;  
 CatTipoDocumento entity = new CatTipoDocumento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoDocumento.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoDocumento.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatTipoDocumento dom = (DominioCatTipoDocumento)obj;  
 CatTipoDocumento entity = new CatTipoDocumento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoDocumento.findById(entity.getTipoDocumentoId()).  
 orElse(new CatTipoDocumento());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoDocumento.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoDocumento.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatTipoDocumento dom = (DominioCatTipoDocumento)obj;  
 CatTipoDocumento entity = new CatTipoDocumento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoDocumento.findById(entity.getTipoDocumentoId()).  
 orElse(new CatTipoDocumento());  
 daoCatTipoDocumento.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoDocumento.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatTipoDocumento dom = (DominioCatTipoDocumento)obj;  
 CatTipoDocumento entity = new CatTipoDocumento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoDocumento.findById(entity.getTipoDocumentoId()).  
 orElse(new CatTipoDocumento());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoDocumento.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatTipoDocumento.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoDocumento.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatTipoDocumento existsById(Object obj) throws Exception {  
 try {  
 DominioCatTipoDocumento dom = (DominioCatTipoDocumento)obj;  
 CatTipoDocumento entity = new CatTipoDocumento();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatTipoDocumento.existsById(entity.getTipoDocumentoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoDocumento.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatTipoFactoresSat.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatTipoFactoresSat;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatTipoFactoresSat;  
import mx.mbi.dgc.jpa.entity.CatTipoFactoresSat;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatTipoFactoresSat implements IfaceCatTipoFactoresSat {  
 @Autowired  
 private DAOCatTipoFactoresSat daoCatTipoFactoresSat;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatTipoFactoresSat> lista = new ArrayList<>();  
 List<CatTipoFactoresSat> objs = daoCatTipoFactoresSat.findAll();  
 for(CatTipoFactoresSat entity: objs) {  
 DominioCatTipoFactoresSat dom = new DominioCatTipoFactoresSat();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoFactoresSat.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatTipoFactoresSat dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatTipoFactoresSat> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatTipoFactoresSat> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatTipoFactoresSat.findByTipoFactorSatId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatTipoFactoresSat.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatTipoFactoresSat.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatTipoFactoresSat entity: objs) {  
 DominioCatTipoFactoresSat to = new DominioCatTipoFactoresSat();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatTipoFactoresSat dom = (DominioCatTipoFactoresSat)obj;  
 CatTipoFactoresSat entity = new CatTipoFactoresSat();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoFactoresSat.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoFactoresSat.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatTipoFactoresSat dom = (DominioCatTipoFactoresSat)obj;  
 CatTipoFactoresSat entity = new CatTipoFactoresSat();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoFactoresSat.findById(entity.getTipoFactorSatId()).  
 orElse(new CatTipoFactoresSat());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoFactoresSat.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoFactoresSat.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatTipoFactoresSat dom = (DominioCatTipoFactoresSat)obj;  
 CatTipoFactoresSat entity = new CatTipoFactoresSat();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoFactoresSat.findById(entity.getTipoFactorSatId()).  
 orElse(new CatTipoFactoresSat());  
 daoCatTipoFactoresSat.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoFactoresSat.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatTipoFactoresSat dom = (DominioCatTipoFactoresSat)obj;  
 CatTipoFactoresSat entity = new CatTipoFactoresSat();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoFactoresSat.findById(entity.getTipoFactorSatId()).  
 orElse(new CatTipoFactoresSat());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoFactoresSat.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatTipoFactoresSat.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoFactoresSat.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatTipoFactoresSat existsById(Object obj) throws Exception {  
 try {  
 DominioCatTipoFactoresSat dom = (DominioCatTipoFactoresSat)obj;  
 CatTipoFactoresSat entity = new CatTipoFactoresSat();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatTipoFactoresSat.existsById(entity.getTipoFactorSatId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoFactoresSat.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatTipoImpuestos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatTipoImpuestos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatTipoImpuestos;  
import mx.mbi.dgc.jpa.entity.CatTipoImpuestos;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatTipoImpuestos implements IfaceCatTipoImpuestos {  
 @Autowired  
 private DAOCatTipoImpuestos daoCatTipoImpuestos;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatTipoImpuestos> lista = new ArrayList<>();  
 List<CatTipoImpuestos> objs = daoCatTipoImpuestos.findAll();  
 for(CatTipoImpuestos entity: objs) {  
 DominioCatTipoImpuestos dom = new DominioCatTipoImpuestos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoImpuestos.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatTipoImpuestos dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatTipoImpuestos> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatTipoImpuestos> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatTipoImpuestos.findByTipoImpuestoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatTipoImpuestos.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatTipoImpuestos.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatTipoImpuestos entity: objs) {  
 DominioCatTipoImpuestos to = new DominioCatTipoImpuestos();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatTipoImpuestos dom = (DominioCatTipoImpuestos)obj;  
 CatTipoImpuestos entity = new CatTipoImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoImpuestos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoImpuestos.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatTipoImpuestos dom = (DominioCatTipoImpuestos)obj;  
 CatTipoImpuestos entity = new CatTipoImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoImpuestos.findById(entity.getTipoImpuestoId()).  
 orElse(new CatTipoImpuestos());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoImpuestos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoImpuestos.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatTipoImpuestos dom = (DominioCatTipoImpuestos)obj;  
 CatTipoImpuestos entity = new CatTipoImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoImpuestos.findById(entity.getTipoImpuestoId()).  
 orElse(new CatTipoImpuestos());  
 daoCatTipoImpuestos.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoImpuestos.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatTipoImpuestos dom = (DominioCatTipoImpuestos)obj;  
 CatTipoImpuestos entity = new CatTipoImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoImpuestos.findById(entity.getTipoImpuestoId()).  
 orElse(new CatTipoImpuestos());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoImpuestos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatTipoImpuestos.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoImpuestos.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatTipoImpuestos existsById(Object obj) throws Exception {  
 try {  
 DominioCatTipoImpuestos dom = (DominioCatTipoImpuestos)obj;  
 CatTipoImpuestos entity = new CatTipoImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatTipoImpuestos.existsById(entity.getTipoImpuestoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoImpuestos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatTipoRelacion.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatTipoRelacion;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatTipoRelacion;  
import mx.mbi.dgc.jpa.entity.CatTipoRelacion;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatTipoRelacion implements IfaceCatTipoRelacion {  
 @Autowired  
 private DAOCatTipoRelacion daoCatTipoRelacion;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatTipoRelacion> lista = new ArrayList<>();  
 List<CatTipoRelacion> objs = daoCatTipoRelacion.findAll();  
 for(CatTipoRelacion entity: objs) {  
 DominioCatTipoRelacion dom = new DominioCatTipoRelacion();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoRelacion.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatTipoRelacion dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatTipoRelacion> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatTipoRelacion> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatTipoRelacion.findByTipoRelacionId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatTipoRelacion.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatTipoRelacion.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatTipoRelacion entity: objs) {  
 DominioCatTipoRelacion to = new DominioCatTipoRelacion();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatTipoRelacion dom = (DominioCatTipoRelacion)obj;  
 CatTipoRelacion entity = new CatTipoRelacion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoRelacion.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoRelacion.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatTipoRelacion dom = (DominioCatTipoRelacion)obj;  
 CatTipoRelacion entity = new CatTipoRelacion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoRelacion.findById(entity.getTipoRelacionId()).  
 orElse(new CatTipoRelacion());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoRelacion.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoRelacion.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatTipoRelacion dom = (DominioCatTipoRelacion)obj;  
 CatTipoRelacion entity = new CatTipoRelacion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoRelacion.findById(entity.getTipoRelacionId()).  
 orElse(new CatTipoRelacion());  
 daoCatTipoRelacion.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoRelacion.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatTipoRelacion dom = (DominioCatTipoRelacion)obj;  
 CatTipoRelacion entity = new CatTipoRelacion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTipoRelacion.findById(entity.getTipoRelacionId()).  
 orElse(new CatTipoRelacion());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoRelacion.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatTipoRelacion.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoRelacion.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatTipoRelacion existsById(Object obj) throws Exception {  
 try {  
 DominioCatTipoRelacion dom = (DominioCatTipoRelacion)obj;  
 CatTipoRelacion entity = new CatTipoRelacion();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatTipoRelacion.existsById(entity.getTipoRelacionId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTipoRelacion.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatTiposDeEvento.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatTiposDeEvento;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatTiposDeEvento;  
import mx.mbi.dgc.jpa.entity.CatTiposDeEvento;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatTiposDeEvento implements IfaceCatTiposDeEvento {  
 @Autowired  
 private DAOCatTiposDeEvento daoCatTiposDeEvento;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatTiposDeEvento> lista = new ArrayList<>();  
 List<CatTiposDeEvento> objs = daoCatTiposDeEvento.findAll();  
 for(CatTiposDeEvento entity: objs) {  
 DominioCatTiposDeEvento dom = new DominioCatTiposDeEvento();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposDeEvento.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatTiposDeEvento dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatTiposDeEvento> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatTiposDeEvento> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatTiposDeEvento.findByTiposDeEventoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatTiposDeEvento.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatTiposDeEvento.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatTiposDeEvento entity: objs) {  
 DominioCatTiposDeEvento to = new DominioCatTiposDeEvento();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatTiposDeEvento dom = (DominioCatTiposDeEvento)obj;  
 CatTiposDeEvento entity = new CatTiposDeEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTiposDeEvento.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposDeEvento.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatTiposDeEvento dom = (DominioCatTiposDeEvento)obj;  
 CatTiposDeEvento entity = new CatTiposDeEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTiposDeEvento.findById(entity.getTiposDeEventoId()).  
 orElse(new CatTiposDeEvento());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTiposDeEvento.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposDeEvento.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatTiposDeEvento dom = (DominioCatTiposDeEvento)obj;  
 CatTiposDeEvento entity = new CatTiposDeEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTiposDeEvento.findById(entity.getTiposDeEventoId()).  
 orElse(new CatTiposDeEvento());  
 daoCatTiposDeEvento.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposDeEvento.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatTiposDeEvento dom = (DominioCatTiposDeEvento)obj;  
 CatTiposDeEvento entity = new CatTiposDeEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTiposDeEvento.findById(entity.getTiposDeEventoId()).  
 orElse(new CatTiposDeEvento());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposDeEvento.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatTiposDeEvento.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposDeEvento.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatTiposDeEvento existsById(Object obj) throws Exception {  
 try {  
 DominioCatTiposDeEvento dom = (DominioCatTiposDeEvento)obj;  
 CatTiposDeEvento entity = new CatTiposDeEvento();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatTiposDeEvento.existsById(entity.getTiposDeEventoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposDeEvento.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatTiposFacturas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatTiposFacturas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatTiposFacturas;  
import mx.mbi.dgc.jpa.entity.CatTiposFacturas;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatTiposFacturas implements IfaceCatTiposFacturas {  
 @Autowired  
 private DAOCatTiposFacturas daoCatTiposFacturas;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatTiposFacturas> lista = new ArrayList<>();  
 List<CatTiposFacturas> objs = daoCatTiposFacturas.findAll();  
 for(CatTiposFacturas entity: objs) {  
 DominioCatTiposFacturas dom = new DominioCatTiposFacturas();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposFacturas.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatTiposFacturas dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatTiposFacturas> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatTiposFacturas> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatTiposFacturas.findByTipoFacturaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatTiposFacturas.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatTiposFacturas.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatTiposFacturas entity: objs) {  
 DominioCatTiposFacturas to = new DominioCatTiposFacturas();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatTiposFacturas dom = (DominioCatTiposFacturas)obj;  
 CatTiposFacturas entity = new CatTiposFacturas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTiposFacturas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposFacturas.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatTiposFacturas dom = (DominioCatTiposFacturas)obj;  
 CatTiposFacturas entity = new CatTiposFacturas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTiposFacturas.findById(entity.getTipoFacturaId()).  
 orElse(new CatTiposFacturas());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTiposFacturas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposFacturas.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatTiposFacturas dom = (DominioCatTiposFacturas)obj;  
 CatTiposFacturas entity = new CatTiposFacturas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTiposFacturas.findById(entity.getTipoFacturaId()).  
 orElse(new CatTiposFacturas());  
 daoCatTiposFacturas.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposFacturas.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatTiposFacturas dom = (DominioCatTiposFacturas)obj;  
 CatTiposFacturas entity = new CatTiposFacturas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTiposFacturas.findById(entity.getTipoFacturaId()).  
 orElse(new CatTiposFacturas());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposFacturas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatTiposFacturas.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposFacturas.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatTiposFacturas existsById(Object obj) throws Exception {  
 try {  
 DominioCatTiposFacturas dom = (DominioCatTiposFacturas)obj;  
 CatTiposFacturas entity = new CatTiposFacturas();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatTiposFacturas.existsById(entity.getTipoFacturaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposFacturas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatTiposServicios.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatTiposServicios;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatTiposServicios;  
import mx.mbi.dgc.jpa.entity.CatTiposServicios;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatTiposServicios implements IfaceCatTiposServicios {  
 @Autowired  
 private DAOCatTiposServicios daoCatTiposServicios;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatTiposServicios> lista = new ArrayList<>();  
 List<CatTiposServicios> objs = daoCatTiposServicios.findAll();  
 for(CatTiposServicios entity: objs) {  
 DominioCatTiposServicios dom = new DominioCatTiposServicios();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposServicios.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatTiposServicios dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatTiposServicios> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatTiposServicios> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatTiposServicios.findByTipoServicioId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatTiposServicios.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatTiposServicios.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatTiposServicios entity: objs) {  
 DominioCatTiposServicios to = new DominioCatTiposServicios();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatTiposServicios dom = (DominioCatTiposServicios)obj;  
 CatTiposServicios entity = new CatTiposServicios();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTiposServicios.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposServicios.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatTiposServicios dom = (DominioCatTiposServicios)obj;  
 CatTiposServicios entity = new CatTiposServicios();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTiposServicios.findById(entity.getTipoServicioId()).  
 orElse(new CatTiposServicios());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTiposServicios.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposServicios.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatTiposServicios dom = (DominioCatTiposServicios)obj;  
 CatTiposServicios entity = new CatTiposServicios();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTiposServicios.findById(entity.getTipoServicioId()).  
 orElse(new CatTiposServicios());  
 daoCatTiposServicios.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposServicios.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatTiposServicios dom = (DominioCatTiposServicios)obj;  
 CatTiposServicios entity = new CatTiposServicios();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatTiposServicios.findById(entity.getTipoServicioId()).  
 orElse(new CatTiposServicios());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposServicios.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatTiposServicios.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposServicios.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatTiposServicios existsById(Object obj) throws Exception {  
 try {  
 DominioCatTiposServicios dom = (DominioCatTiposServicios)obj;  
 CatTiposServicios entity = new CatTiposServicios();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatTiposServicios.existsById(entity.getTipoServicioId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatTiposServicios.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatUnidad.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatUnidad;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatUnidad;  
import mx.mbi.dgc.jpa.entity.CatUnidad;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatUnidad implements IfaceCatUnidad {  
 @Autowired  
 private DAOCatUnidad daoCatUnidad;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatUnidad> lista = new ArrayList<>();  
 List<CatUnidad> objs = daoCatUnidad.findAll();  
 for(CatUnidad entity: objs) {  
 DominioCatUnidad dom = new DominioCatUnidad();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUnidad.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatUnidad dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatUnidad> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatUnidad> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatUnidad.findByUnidadId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatUnidad.findByUnidadContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatUnidad.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatUnidad entity: objs) {  
 DominioCatUnidad to = new DominioCatUnidad();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatUnidad dom = (DominioCatUnidad)obj;  
 CatUnidad entity = new CatUnidad();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatUnidad.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUnidad.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatUnidad dom = (DominioCatUnidad)obj;  
 CatUnidad entity = new CatUnidad();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatUnidad.findById(entity.getUnidadId()).  
 orElse(new CatUnidad());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatUnidad.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUnidad.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatUnidad dom = (DominioCatUnidad)obj;  
 CatUnidad entity = new CatUnidad();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatUnidad.findById(entity.getUnidadId()).  
 orElse(new CatUnidad());  
 daoCatUnidad.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUnidad.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatUnidad dom = (DominioCatUnidad)obj;  
 CatUnidad entity = new CatUnidad();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatUnidad.findById(entity.getUnidadId()).  
 orElse(new CatUnidad());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUnidad.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatUnidad.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUnidad.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatUnidad existsById(Object obj) throws Exception {  
 try {  
 DominioCatUnidad dom = (DominioCatUnidad)obj;  
 CatUnidad entity = new CatUnidad();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatUnidad.existsById(entity.getUnidadId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUnidad.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatUnidades.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatUnidades ;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatUnidades ;  
import mx.mbi.dgc.jpa.entity.CatUnidades ;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatUnidades implements IfaceCatUnidades {  
 @Autowired  
 private DAOCatUnidades daoCatUnidades ;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatUnidades > lista = new ArrayList<>();  
 List<CatUnidades > objs = daoCatUnidades .findAll();  
 for(CatUnidades entity: objs) {  
 DominioCatUnidades dom = new DominioCatUnidades ();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUnidades .findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatUnidades dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatUnidades > lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatUnidades > objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatUnidades .findByUnidadId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatUnidades .findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatUnidades .findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatUnidades entity: objs) {  
 DominioCatUnidades to = new DominioCatUnidades ();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatUnidades dom = (DominioCatUnidades )obj;  
 CatUnidades entity = new CatUnidades ();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatUnidades .save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUnidades .save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatUnidades dom = (DominioCatUnidades )obj;  
 CatUnidades entity = new CatUnidades ();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatUnidades .findById(entity.getUnidadId()).  
 orElse(new CatUnidades ());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatUnidades .save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUnidades .update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatUnidades dom = (DominioCatUnidades )obj;  
 CatUnidades entity = new CatUnidades ();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatUnidades .findById(entity.getUnidadId()).  
 orElse(new CatUnidades ());  
 daoCatUnidades .delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUnidades .delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatUnidades dom = (DominioCatUnidades )obj;  
 CatUnidades entity = new CatUnidades ();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatUnidades .findById(entity.getUnidadId()).  
 orElse(new CatUnidades ());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUnidades .findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatUnidades .count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUnidades .count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatUnidades existsById(Object obj) throws Exception {  
 try {  
 DominioCatUnidades dom = (DominioCatUnidades )obj;  
 CatUnidades entity = new CatUnidades ();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatUnidades .existsById(entity.getUnidadId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUnidades .findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatUsoCfdi.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatUsoCfdi;  
import mx.mbi.dgc.dominio.DominioCatUsoCfdi;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.jpa.entity.CatUsoCfdi;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCatUsoCfdi implements IfaceCatUsoCfdi {  
 @Autowired  
 private DAOCatUsoCfdi daoCatUsoCfdi;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatUsoCfdi> lista = new ArrayList<>();  
 List<CatUsoCfdi> objs = daoCatUsoCfdi.findAll();  
 for(CatUsoCfdi entity: objs) {  
 DominioCatUsoCfdi dom = new DominioCatUsoCfdi();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUsoCfdi.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatUsoCfdi dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatUsoCfdi> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatUsoCfdi> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatUsoCfdi.findByUsoCfdiId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatUsoCfdi.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatUsoCfdi.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatUsoCfdi entity: objs) {  
 DominioCatUsoCfdi to = new DominioCatUsoCfdi();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatUsoCfdi dom = (DominioCatUsoCfdi)obj;  
 CatUsoCfdi entity = new CatUsoCfdi();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatUsoCfdi.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUsoCfdi.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatUsoCfdi dom = (DominioCatUsoCfdi)obj;  
 CatUsoCfdi entity = new CatUsoCfdi();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatUsoCfdi.findById(entity.getUsoCfdiId()).  
 orElse(new CatUsoCfdi());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatUsoCfdi.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUsoCfdi.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatUsoCfdi dom = (DominioCatUsoCfdi)obj;  
 CatUsoCfdi entity = new CatUsoCfdi();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatUsoCfdi.findById(entity.getUsoCfdiId()).  
 orElse(new CatUsoCfdi());  
 daoCatUsoCfdi.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUsoCfdi.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatUsoCfdi dom = (DominioCatUsoCfdi)obj;  
 CatUsoCfdi entity = new CatUsoCfdi();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatUsoCfdi.findById(entity.getUsoCfdiId()).  
 orElse(new CatUsoCfdi());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUsoCfdi.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatUsoCfdi.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUsoCfdi.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatUsoCfdi existsById(Object obj) throws Exception {  
 try {  
 DominioCatUsoCfdi dom = (DominioCatUsoCfdi)obj;  
 CatUsoCfdi entity = new CatUsoCfdi();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatUsoCfdi.existsById(entity.getUsoCfdiId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatUsoCfdi.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatVariable.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatVariable;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatVariable;  
import mx.mbi.dgc.jpa.entity.CatVariable;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatVariable implements IfaceCatVariable {  
 @Autowired  
 private DAOCatVariable daoCatVariable;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatVariable> lista = new ArrayList<>();  
 List<CatVariable> objs = daoCatVariable.findAll();  
 for(CatVariable entity: objs) {  
 DominioCatVariable dom = new DominioCatVariable();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVariable.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatVariable dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatVariable> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatVariable> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatVariable.findByVariableId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatVariable.findByUnidadMedidaContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatVariable.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatVariable entity: objs) {  
 DominioCatVariable to = new DominioCatVariable();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatVariable dom = (DominioCatVariable)obj;  
 CatVariable entity = new CatVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatVariable.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVariable.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatVariable dom = (DominioCatVariable)obj;  
 CatVariable entity = new CatVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatVariable.findById(entity.getVariableId()).  
 orElse(new CatVariable());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatVariable.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVariable.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatVariable dom = (DominioCatVariable)obj;  
 CatVariable entity = new CatVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatVariable.findById(entity.getVariableId()).  
 orElse(new CatVariable());  
 daoCatVariable.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVariable.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatVariable dom = (DominioCatVariable)obj;  
 CatVariable entity = new CatVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatVariable.findById(entity.getVariableId()).  
 orElse(new CatVariable());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVariable.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatVariable.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVariable.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatVariable existsById(Object obj) throws Exception {  
 try {  
 DominioCatVariable dom = (DominioCatVariable)obj;  
 CatVariable entity = new CatVariable();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatVariable.existsById(entity.getVariableId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVariable.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatVariablePerfil.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatVariablePerfil;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatVariablePerfil;  
import mx.mbi.dgc.jpa.entity.CatVariablePerfil;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatVariablePerfil implements IfaceCatVariablePerfil {  
 @Autowired  
 private DAOCatVariablePerfil daoCatVariablePerfil;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatVariablePerfil> lista = new ArrayList<>();  
 List<CatVariablePerfil> objs = daoCatVariablePerfil.findAll();  
 for(CatVariablePerfil entity: objs) {  
 DominioCatVariablePerfil dom = new DominioCatVariablePerfil();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVariablePerfil.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatVariablePerfil dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatVariablePerfil> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatVariablePerfil> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatVariablePerfil.findByVariablePerfilId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatVariablePerfil.findByVariablePerfilContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatVariablePerfil.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatVariablePerfil entity: objs) {  
 DominioCatVariablePerfil to = new DominioCatVariablePerfil();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatVariablePerfil dom = (DominioCatVariablePerfil)obj;  
 CatVariablePerfil entity = new CatVariablePerfil();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatVariablePerfil.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVariablePerfil.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatVariablePerfil dom = (DominioCatVariablePerfil)obj;  
 CatVariablePerfil entity = new CatVariablePerfil();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatVariablePerfil.findById(entity.getVariablePerfilId()).  
 orElse(new CatVariablePerfil());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatVariablePerfil.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVariablePerfil.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatVariablePerfil dom = (DominioCatVariablePerfil)obj;  
 CatVariablePerfil entity = new CatVariablePerfil();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatVariablePerfil.findById(entity.getVariablePerfilId()).  
 orElse(new CatVariablePerfil());  
 daoCatVariablePerfil.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVariablePerfil.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatVariablePerfil dom = (DominioCatVariablePerfil)obj;  
 CatVariablePerfil entity = new CatVariablePerfil();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatVariablePerfil.findById(entity.getVariablePerfilId()).  
 orElse(new CatVariablePerfil());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVariablePerfil.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatVariablePerfil.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVariablePerfil.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatVariablePerfil existsById(Object obj) throws Exception {  
 try {  
 DominioCatVariablePerfil dom = (DominioCatVariablePerfil)obj;  
 CatVariablePerfil entity = new CatVariablePerfil();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatVariablePerfil.existsById(entity.getVariablePerfilId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVariablePerfil.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessCatVersion.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatVersion;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOCatVersion;  
import mx.mbi.dgc.jpa.entity.CatVersion;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessCatVersion implements IfaceCatVersion {  
 @Autowired  
 private DAOCatVersion daoCatVersion;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioCatVersion> lista = new ArrayList<>();  
 List<CatVersion> objs = daoCatVersion.findAll();  
 for(CatVersion entity: objs) {  
 DominioCatVersion dom = new DominioCatVersion();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVersion.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioCatVersion dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioCatVersion> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<CatVersion> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoCatVersion.findByVersionId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoCatVersion.findByVersionContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoCatVersion.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(CatVersion entity: objs) {  
 DominioCatVersion to = new DominioCatVersion();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioCatVersion dom = (DominioCatVersion)obj;  
 CatVersion entity = new CatVersion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatVersion.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVersion.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioCatVersion dom = (DominioCatVersion)obj;  
 CatVersion entity = new CatVersion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatVersion.findById(entity.getVersionId()).  
 orElse(new CatVersion());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatVersion.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVersion.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioCatVersion dom = (DominioCatVersion)obj;  
 CatVersion entity = new CatVersion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatVersion.findById(entity.getVersionId()).  
 orElse(new CatVersion());  
 daoCatVersion.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVersion.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioCatVersion dom = (DominioCatVersion)obj;  
 CatVersion entity = new CatVersion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoCatVersion.findById(entity.getVersionId()).  
 orElse(new CatVersion());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVersion.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoCatVersion.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVersion.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioCatVersion existsById(Object obj) throws Exception {  
 try {  
 DominioCatVersion dom = (DominioCatVersion)obj;  
 CatVersion entity = new CatVersion();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoCatVersion.existsById(entity.getVersionId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatVersion.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessLogin.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.negocio.DAOLogin;  
import mx.mbi.dgc.dao.negocio.DaoTxUsuarioCatPlantas;  
import mx.mbi.dgc.dominio.DominioCatPlanta;  
import mx.mbi.dgc.dominio.DominioTxUsuario;  
import mx.mbi.dgc.dominio.DominioTxUsuarioPlanta;  
import mx.mbi.dgc.jpa.entity.TxUsuario;  
import mx.mbi.dgc.jpa.entity.TxUsuarioPlanta;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessLogin implements IfaceLogin {  
 @Autowired  
 private DAOLogin daoLogin;  
   
 @Autowired  
 private DaoTxUsuarioCatPlantas daoTxUsuarioCatPlantas;  
  
 @Override  
 public DominioTxUsuario findByUsername(DominioTxUsuario dom) throws Exception {  
 try {  
 TxUsuario entity = new TxUsuario();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoLogin.findByUsername(dom.getUsername());  
 List<DominioTxUsuarioPlanta> txUsuarioPlantas = new ArrayList<>();  
 DominioTxUsuarioPlanta dominioTxUsuarioPlanta = null;  
 DominioCatPlanta DominioCatPlanta = null;  
 List<TxUsuarioPlanta> listaPlantas = daoTxUsuarioCatPlantas.getListaPlantas(entity.getUsuarioId());  
 for(TxUsuarioPlanta txUsuarioPlanta : listaPlantas){  
 dominioTxUsuarioPlanta = new DominioTxUsuarioPlanta();  
 DominioCatPlanta dominioCatPlanta = new DominioCatPlanta();  
 dominioCatPlanta.setNombre(txUsuarioPlanta.getCatPlanta().getNombre());  
 dominioCatPlanta.setPlantaId(txUsuarioPlanta.getCatPlanta().getPlantaId());  
 dominioTxUsuarioPlanta.setCatPlanta(dominioCatPlanta);  
 txUsuarioPlantas.add(dominioTxUsuarioPlanta);  
 }  
 dom.setTxUsuarioPlantas(txUsuarioPlantas);  
 dom.setFound(false);  
 if(entity != null) {  
 JRUtil.copyProperties(dom, entity);  
 dom.setCode(200);  
 dom.setStatus(true);  
 dom.setFound(true);  
 }  
 return dom;  
 } catch(Exception ex) {  
 ex.printStackTrace();  
 throw new Exception("Error in BusinessTxUsuario.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAll() throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 // TODO Implement this method  
 return null;  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 // TODO Implement this method  
 return null;  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 // TODO Implement this method  
 return null;  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 // TODO Implement this method  
 return null;  
 }  
  
 @Override  
 public long count() throws Exception {  
 // TODO Implement this method  
 return 0L;  
 }  
  
 @Override  
 public Object existsById(Object obj) throws Exception {  
 // TODO Implement this method  
 return null;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessSatImpuestos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioSatImpuestos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOSatImpuestos;  
import mx.mbi.dgc.jpa.entity.SatImpuestos;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessSatImpuestos implements IfaceSatImpuestos {  
 @Autowired  
 private DAOSatImpuestos daoSatImpuestos;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioSatImpuestos> lista = new ArrayList<>();  
 List<SatImpuestos> objs = daoSatImpuestos.findAll();  
 for(SatImpuestos entity: objs) {  
 DominioSatImpuestos dom = new DominioSatImpuestos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatImpuestos.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioSatImpuestos dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioSatImpuestos> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<SatImpuestos> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoSatImpuestos.findByImpuestoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoSatImpuestos.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoSatImpuestos.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(SatImpuestos entity: objs) {  
 DominioSatImpuestos to = new DominioSatImpuestos();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioSatImpuestos dom = (DominioSatImpuestos)obj;  
 SatImpuestos entity = new SatImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatImpuestos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatImpuestos.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioSatImpuestos dom = (DominioSatImpuestos)obj;  
 SatImpuestos entity = new SatImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatImpuestos.findById(entity.getImpuestoId()).  
 orElse(new SatImpuestos());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatImpuestos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatImpuestos.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioSatImpuestos dom = (DominioSatImpuestos)obj;  
 SatImpuestos entity = new SatImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatImpuestos.findById(entity.getImpuestoId()).  
 orElse(new SatImpuestos());  
 daoSatImpuestos.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatImpuestos.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioSatImpuestos dom = (DominioSatImpuestos)obj;  
 SatImpuestos entity = new SatImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatImpuestos.findById(entity.getImpuestoId()).  
 orElse(new SatImpuestos());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatImpuestos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoSatImpuestos.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatImpuestos.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioSatImpuestos existsById(Object obj) throws Exception {  
 try {  
 DominioSatImpuestos dom = (DominioSatImpuestos)obj;  
 SatImpuestos entity = new SatImpuestos();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoSatImpuestos.existsById(entity.getImpuestoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatImpuestos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessSatMonedas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioSatMonedas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOSatMonedas;  
import mx.mbi.dgc.jpa.entity.SatMonedas;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessSatMonedas implements IfaceSatMonedas {  
 @Autowired  
 private DAOSatMonedas daoSatMonedas;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioSatMonedas> lista = new ArrayList<>();  
 List<SatMonedas> objs = daoSatMonedas.findAll();  
 for(SatMonedas entity: objs) {  
 DominioSatMonedas dom = new DominioSatMonedas();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatMonedas.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioSatMonedas dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioSatMonedas> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<SatMonedas> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoSatMonedas.findByMonedaSatId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoSatMonedas.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoSatMonedas.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(SatMonedas entity: objs) {  
 DominioSatMonedas to = new DominioSatMonedas();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioSatMonedas dom = (DominioSatMonedas)obj;  
 SatMonedas entity = new SatMonedas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatMonedas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatMonedas.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioSatMonedas dom = (DominioSatMonedas)obj;  
 SatMonedas entity = new SatMonedas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatMonedas.findById(entity.getMonedaSatId()).  
 orElse(new SatMonedas());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatMonedas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatMonedas.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioSatMonedas dom = (DominioSatMonedas)obj;  
 SatMonedas entity = new SatMonedas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatMonedas.findById(entity.getMonedaSatId()).  
 orElse(new SatMonedas());  
 daoSatMonedas.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatMonedas.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioSatMonedas dom = (DominioSatMonedas)obj;  
 SatMonedas entity = new SatMonedas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatMonedas.findById(entity.getMonedaSatId()).  
 orElse(new SatMonedas());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatMonedas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoSatMonedas.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatMonedas.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioSatMonedas existsById(Object obj) throws Exception {  
 try {  
 DominioSatMonedas dom = (DominioSatMonedas)obj;  
 SatMonedas entity = new SatMonedas();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoSatMonedas.existsById(entity.getMonedaSatId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatMonedas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessSatPaises.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioSatPaises;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOSatPaises;  
import mx.mbi.dgc.jpa.entity.SatPaises;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessSatPaises implements IfaceSatPaises {  
 @Autowired  
 private DAOSatPaises daoSatPaises;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioSatPaises> lista = new ArrayList<>();  
 List<SatPaises> objs = daoSatPaises.findAll();  
 for(SatPaises entity: objs) {  
 DominioSatPaises dom = new DominioSatPaises();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatPaises.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioSatPaises dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioSatPaises> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<SatPaises> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoSatPaises.findByPaisSatId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoSatPaises.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoSatPaises.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(SatPaises entity: objs) {  
 DominioSatPaises to = new DominioSatPaises();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioSatPaises dom = (DominioSatPaises)obj;  
 SatPaises entity = new SatPaises();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatPaises.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatPaises.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioSatPaises dom = (DominioSatPaises)obj;  
 SatPaises entity = new SatPaises();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatPaises.findById(entity.getPaisSatId()).  
 orElse(new SatPaises());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatPaises.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatPaises.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioSatPaises dom = (DominioSatPaises)obj;  
 SatPaises entity = new SatPaises();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatPaises.findById(entity.getPaisSatId()).  
 orElse(new SatPaises());  
 daoSatPaises.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatPaises.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioSatPaises dom = (DominioSatPaises)obj;  
 SatPaises entity = new SatPaises();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatPaises.findById(entity.getPaisSatId()).  
 orElse(new SatPaises());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatPaises.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoSatPaises.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatPaises.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioSatPaises existsById(Object obj) throws Exception {  
 try {  
 DominioSatPaises dom = (DominioSatPaises)obj;  
 SatPaises entity = new SatPaises();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoSatPaises.existsById(entity.getPaisSatId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatPaises.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessSatProductosClaves.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioSatProductosClaves;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOSatProductosClaves;  
import mx.mbi.dgc.jpa.entity.SatProductosClaves;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessSatProductosClaves implements IfaceSatProductosClaves {  
 @Autowired  
 private DAOSatProductosClaves daoSatProductosClaves;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioSatProductosClaves> lista = new ArrayList<>();  
 List<SatProductosClaves> objs = daoSatProductosClaves.findAll();  
 for(SatProductosClaves entity: objs) {  
 DominioSatProductosClaves dom = new DominioSatProductosClaves();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatProductosClaves.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioSatProductosClaves dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioSatProductosClaves> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<SatProductosClaves> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoSatProductosClaves.findBySatProductosClavesId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoSatProductosClaves.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoSatProductosClaves.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(SatProductosClaves entity: objs) {  
 DominioSatProductosClaves to = new DominioSatProductosClaves();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioSatProductosClaves dom = (DominioSatProductosClaves)obj;  
 SatProductosClaves entity = new SatProductosClaves();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatProductosClaves.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatProductosClaves.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioSatProductosClaves dom = (DominioSatProductosClaves)obj;  
 SatProductosClaves entity = new SatProductosClaves();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatProductosClaves.findById(entity.getSatProductosClavesId()).  
 orElse(new SatProductosClaves());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatProductosClaves.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatProductosClaves.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioSatProductosClaves dom = (DominioSatProductosClaves)obj;  
 SatProductosClaves entity = new SatProductosClaves();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatProductosClaves.findById(entity.getSatProductosClavesId()).  
 orElse(new SatProductosClaves());  
 daoSatProductosClaves.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatProductosClaves.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioSatProductosClaves dom = (DominioSatProductosClaves)obj;  
 SatProductosClaves entity = new SatProductosClaves();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatProductosClaves.findById(entity.getSatProductosClavesId()).  
 orElse(new SatProductosClaves());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatProductosClaves.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoSatProductosClaves.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatProductosClaves.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioSatProductosClaves existsById(Object obj) throws Exception {  
 try {  
 DominioSatProductosClaves dom = (DominioSatProductosClaves)obj;  
 SatProductosClaves entity = new SatProductosClaves();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoSatProductosClaves.existsById(entity.getSatProductosClavesId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatProductosClaves.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessSatProductosUnidades.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioSatProductosUnidades;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOSatProductosUnidades;  
import mx.mbi.dgc.jpa.entity.SatProductosUnidades;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessSatProductosUnidades implements IfaceSatProductosUnidades {  
 @Autowired  
 private DAOSatProductosUnidades daoSatProductosUnidades;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioSatProductosUnidades> lista = new ArrayList<>();  
 List<SatProductosUnidades> objs = daoSatProductosUnidades.findAll();  
 for(SatProductosUnidades entity: objs) {  
 DominioSatProductosUnidades dom = new DominioSatProductosUnidades();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatProductosUnidades.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioSatProductosUnidades dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioSatProductosUnidades> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<SatProductosUnidades> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoSatProductosUnidades.findBySatProductosUnidadesId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoSatProductosUnidades.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoSatProductosUnidades.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(SatProductosUnidades entity: objs) {  
 DominioSatProductosUnidades to = new DominioSatProductosUnidades();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioSatProductosUnidades dom = (DominioSatProductosUnidades)obj;  
 SatProductosUnidades entity = new SatProductosUnidades();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatProductosUnidades.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatProductosUnidades.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioSatProductosUnidades dom = (DominioSatProductosUnidades)obj;  
 SatProductosUnidades entity = new SatProductosUnidades();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatProductosUnidades.findById(entity.getSatProductosUnidadesId()).  
 orElse(new SatProductosUnidades());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatProductosUnidades.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatProductosUnidades.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioSatProductosUnidades dom = (DominioSatProductosUnidades)obj;  
 SatProductosUnidades entity = new SatProductosUnidades();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatProductosUnidades.findById(entity.getSatProductosUnidadesId()).  
 orElse(new SatProductosUnidades());  
 daoSatProductosUnidades.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatProductosUnidades.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioSatProductosUnidades dom = (DominioSatProductosUnidades)obj;  
 SatProductosUnidades entity = new SatProductosUnidades();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatProductosUnidades.findById(entity.getSatProductosUnidadesId()).  
 orElse(new SatProductosUnidades());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatProductosUnidades.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoSatProductosUnidades.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatProductosUnidades.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioSatProductosUnidades existsById(Object obj) throws Exception {  
 try {  
 DominioSatProductosUnidades dom = (DominioSatProductosUnidades)obj;  
 SatProductosUnidades entity = new SatProductosUnidades();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoSatProductosUnidades.existsById(entity.getSatProductosUnidadesId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatProductosUnidades.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessSatTasaCuota.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioSatTasaCuota;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOSatTasaCuota;  
import mx.mbi.dgc.jpa.entity.SatTasaCuota;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessSatTasaCuota implements IfaceSatTasaCuota {  
 @Autowired  
 private DAOSatTasaCuota daoSatTasaCuota;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioSatTasaCuota> lista = new ArrayList<>();  
 List<SatTasaCuota> objs = daoSatTasaCuota.findAll();  
 for(SatTasaCuota entity: objs) {  
 DominioSatTasaCuota dom = new DominioSatTasaCuota();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTasaCuota.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioSatTasaCuota dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioSatTasaCuota> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<SatTasaCuota> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoSatTasaCuota.findByTasaCuotaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoSatTasaCuota.findByRangoContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoSatTasaCuota.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(SatTasaCuota entity: objs) {  
 DominioSatTasaCuota to = new DominioSatTasaCuota();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioSatTasaCuota dom = (DominioSatTasaCuota)obj;  
 SatTasaCuota entity = new SatTasaCuota();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatTasaCuota.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTasaCuota.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioSatTasaCuota dom = (DominioSatTasaCuota)obj;  
 SatTasaCuota entity = new SatTasaCuota();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatTasaCuota.findById(entity.getTasaCuotaId()).  
 orElse(new SatTasaCuota());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatTasaCuota.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTasaCuota.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioSatTasaCuota dom = (DominioSatTasaCuota)obj;  
 SatTasaCuota entity = new SatTasaCuota();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatTasaCuota.findById(entity.getTasaCuotaId()).  
 orElse(new SatTasaCuota());  
 daoSatTasaCuota.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTasaCuota.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioSatTasaCuota dom = (DominioSatTasaCuota)obj;  
 SatTasaCuota entity = new SatTasaCuota();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatTasaCuota.findById(entity.getTasaCuotaId()).  
 orElse(new SatTasaCuota());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTasaCuota.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoSatTasaCuota.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTasaCuota.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioSatTasaCuota existsById(Object obj) throws Exception {  
 try {  
 DominioSatTasaCuota dom = (DominioSatTasaCuota)obj;  
 SatTasaCuota entity = new SatTasaCuota();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoSatTasaCuota.existsById(entity.getTasaCuotaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTasaCuota.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessSatTipoFactor.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioSatTipoFactor;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOSatTipoFactor;  
import mx.mbi.dgc.jpa.entity.SatTipoFactor;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessSatTipoFactor implements IfaceSatTipoFactor {  
 @Autowired  
 private DAOSatTipoFactor daoSatTipoFactor;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioSatTipoFactor> lista = new ArrayList<>();  
 List<SatTipoFactor> objs = daoSatTipoFactor.findAll();  
 for(SatTipoFactor entity: objs) {  
 DominioSatTipoFactor dom = new DominioSatTipoFactor();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTipoFactor.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioSatTipoFactor dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioSatTipoFactor> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<SatTipoFactor> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoSatTipoFactor.findByTipoFactorId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoSatTipoFactor.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoSatTipoFactor.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(SatTipoFactor entity: objs) {  
 DominioSatTipoFactor to = new DominioSatTipoFactor();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioSatTipoFactor dom = (DominioSatTipoFactor)obj;  
 SatTipoFactor entity = new SatTipoFactor();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatTipoFactor.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTipoFactor.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioSatTipoFactor dom = (DominioSatTipoFactor)obj;  
 SatTipoFactor entity = new SatTipoFactor();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatTipoFactor.findById(entity.getTipoFactorId()).  
 orElse(new SatTipoFactor());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatTipoFactor.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTipoFactor.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioSatTipoFactor dom = (DominioSatTipoFactor)obj;  
 SatTipoFactor entity = new SatTipoFactor();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatTipoFactor.findById(entity.getTipoFactorId()).  
 orElse(new SatTipoFactor());  
 daoSatTipoFactor.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTipoFactor.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioSatTipoFactor dom = (DominioSatTipoFactor)obj;  
 SatTipoFactor entity = new SatTipoFactor();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatTipoFactor.findById(entity.getTipoFactorId()).  
 orElse(new SatTipoFactor());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTipoFactor.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoSatTipoFactor.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTipoFactor.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioSatTipoFactor existsById(Object obj) throws Exception {  
 try {  
 DominioSatTipoFactor dom = (DominioSatTipoFactor)obj;  
 SatTipoFactor entity = new SatTipoFactor();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoSatTipoFactor.existsById(entity.getTipoFactorId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTipoFactor.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessSatTipoRelacion.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioSatTipoRelacion;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOSatTipoRelacion;  
import mx.mbi.dgc.jpa.entity.SatTipoRelacion;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessSatTipoRelacion implements IfaceSatTipoRelacion {  
 @Autowired  
 private DAOSatTipoRelacion daoSatTipoRelacion;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioSatTipoRelacion> lista = new ArrayList<>();  
 List<SatTipoRelacion> objs = daoSatTipoRelacion.findAll();  
 for(SatTipoRelacion entity: objs) {  
 DominioSatTipoRelacion dom = new DominioSatTipoRelacion();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTipoRelacion.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioSatTipoRelacion dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioSatTipoRelacion> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<SatTipoRelacion> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoSatTipoRelacion.findByNombre(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoSatTipoRelacion.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoSatTipoRelacion.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(SatTipoRelacion entity: objs) {  
 DominioSatTipoRelacion to = new DominioSatTipoRelacion();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioSatTipoRelacion dom = (DominioSatTipoRelacion)obj;  
 SatTipoRelacion entity = new SatTipoRelacion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatTipoRelacion.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTipoRelacion.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioSatTipoRelacion dom = (DominioSatTipoRelacion)obj;  
 SatTipoRelacion entity = new SatTipoRelacion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatTipoRelacion.findById(entity.getNombre()).  
 orElse(new SatTipoRelacion());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatTipoRelacion.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTipoRelacion.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioSatTipoRelacion dom = (DominioSatTipoRelacion)obj;  
 SatTipoRelacion entity = new SatTipoRelacion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatTipoRelacion.findById(entity.getNombre()).  
 orElse(new SatTipoRelacion());  
 daoSatTipoRelacion.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTipoRelacion.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioSatTipoRelacion dom = (DominioSatTipoRelacion)obj;  
 SatTipoRelacion entity = new SatTipoRelacion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatTipoRelacion.findById(entity.getNombre()).  
 orElse(new SatTipoRelacion());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTipoRelacion.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoSatTipoRelacion.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTipoRelacion.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioSatTipoRelacion existsById(Object obj) throws Exception {  
 try {  
 DominioSatTipoRelacion dom = (DominioSatTipoRelacion)obj;  
 SatTipoRelacion entity = new SatTipoRelacion();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoSatTipoRelacion.existsById(entity.getNombre()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatTipoRelacion.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessSatUsoCfdi.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioSatUsoCfdi;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOSatUsoCfdi;  
import mx.mbi.dgc.jpa.entity.SatUsoCfdi;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessSatUsoCfdi implements IfaceSatUsoCfdi {  
 @Autowired  
 private DAOSatUsoCfdi daoSatUsoCfdi;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioSatUsoCfdi> lista = new ArrayList<>();  
 List<SatUsoCfdi> objs = daoSatUsoCfdi.findAll();  
 for(SatUsoCfdi entity: objs) {  
 DominioSatUsoCfdi dom = new DominioSatUsoCfdi();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatUsoCfdi.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioSatUsoCfdi dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioSatUsoCfdi> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<SatUsoCfdi> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoSatUsoCfdi.findByDescripcion(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoSatUsoCfdi.findByDescripcionContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoSatUsoCfdi.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(SatUsoCfdi entity: objs) {  
 DominioSatUsoCfdi to = new DominioSatUsoCfdi();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioSatUsoCfdi dom = (DominioSatUsoCfdi)obj;  
 SatUsoCfdi entity = new SatUsoCfdi();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatUsoCfdi.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatUsoCfdi.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioSatUsoCfdi dom = (DominioSatUsoCfdi)obj;  
 SatUsoCfdi entity = new SatUsoCfdi();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatUsoCfdi.findById(entity.getDescripcion()).  
 orElse(new SatUsoCfdi());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatUsoCfdi.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatUsoCfdi.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioSatUsoCfdi dom = (DominioSatUsoCfdi)obj;  
 SatUsoCfdi entity = new SatUsoCfdi();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatUsoCfdi.findById(entity.getDescripcion()).  
 orElse(new SatUsoCfdi());  
 daoSatUsoCfdi.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatUsoCfdi.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioSatUsoCfdi dom = (DominioSatUsoCfdi)obj;  
 SatUsoCfdi entity = new SatUsoCfdi();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoSatUsoCfdi.findById(entity.getDescripcion()).  
 orElse(new SatUsoCfdi());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatUsoCfdi.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoSatUsoCfdi.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatUsoCfdi.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioSatUsoCfdi existsById(Object obj) throws Exception {  
 try {  
 DominioSatUsoCfdi dom = (DominioSatUsoCfdi)obj;  
 SatUsoCfdi entity = new SatUsoCfdi();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoSatUsoCfdi.existsById(entity.getDescripcion()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessSatUsoCfdi.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxAdjuntos.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOTxAdjuntos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxAdjuntos;  
import mx.mbi.dgc.dominio.DominioTxEvento;  
import mx.mbi.dgc.jpa.entity.TxAdjuntos;  
import mx.mbi.dgc.jpa.entity.TxEvento;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessTxAdjuntos implements IfaceTxAdjuntos {  
 @Autowired  
 private DAOTxAdjuntos daoTxAdjuntos;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxAdjuntos> lista = new ArrayList<>();  
 List<TxAdjuntos> objs = daoTxAdjuntos.findAll();  
 for (TxAdjuntos entity : objs) {  
 DominioTxAdjuntos dom = new DominioTxAdjuntos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxAdjuntos.findAll: " + ex);  
 }  
 }  
  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxAdjuntos dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxAdjuntos> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxAdjuntos> objs = null;  
 if (dom.getSearch() != null && dom.getSearch().getValue() != null &&  
 ("" + dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs =  
 daoTxAdjuntos.findByAdjuntoId(new java.math.BigDecimal("" + dom.getSearch().getValue()),  
 PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC : Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex) {  
 try {  
 objs =  
 daoTxAdjuntos.findByReferenciaContaining("" + dom.getSearch().getValue(),  
 PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC :  
 Sort.Direction.DESC,  
 dom.getCampos()  
 .get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex2) {  
 }  
 }  
 } else {  
 objs =  
 daoTxAdjuntos.findAll(PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC : Sort.Direction.DESC,  
 dom.getCampos()  
 .get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 }  
 for (TxAdjuntos entity : objs) {  
 DominioTxAdjuntos to = new DominioTxAdjuntos();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: " + ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxAdjuntos dom = (DominioTxAdjuntos) obj;  
 TxAdjuntos entity = new TxAdjuntos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxAdjuntos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxAdjuntos.save: " + ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxAdjuntos dom = (DominioTxAdjuntos) obj;  
 TxAdjuntos entity = new TxAdjuntos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxAdjuntos.findById(entity.getAdjuntoId()).orElse(new TxAdjuntos());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxAdjuntos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxAdjuntos.update: " + ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxAdjuntos dom = (DominioTxAdjuntos) obj;  
 TxAdjuntos entity = new TxAdjuntos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxAdjuntos.findById(entity.getAdjuntoId()).orElse(new TxAdjuntos());  
 daoTxAdjuntos.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxAdjuntos.delete: " + ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxAdjuntos dom = (DominioTxAdjuntos) obj;  
 TxAdjuntos entity = new TxAdjuntos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxAdjuntos.findById(entity.getAdjuntoId()).orElse(new TxAdjuntos());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxAdjuntos.findById: " + ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxAdjuntos.count();  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxAdjuntos.count: " + ex);  
 }  
 }  
  
 @Override  
 public DominioTxAdjuntos existsById(Object obj) throws Exception {  
 try {  
 DominioTxAdjuntos dom = (DominioTxAdjuntos) obj;  
 TxAdjuntos entity = new TxAdjuntos();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxAdjuntos.existsById(entity.getAdjuntoId()));  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxAdjuntos.findById: " + ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
  
 @Override  
 public List<DominioTxAdjuntos> findByEventoId(DominioTxEvento evento) throws Exception {  
 try {  
 TxEvento entityEvento = new TxEvento();  
 JRUtil.copyProperties(entityEvento, evento);  
 List<DominioTxAdjuntos> lista = new ArrayList<>();  
 List<TxAdjuntos> objs = daoTxAdjuntos.findByEventoId(entityEvento);  
 for (TxAdjuntos entity : objs) {  
 DominioTxAdjuntos dom = new DominioTxAdjuntos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 return null;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxBitacora.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxBitacora;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxBitacora;  
import mx.mbi.dgc.jpa.entity.TxBitacora;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxBitacora implements IfaceTxBitacora {  
 @Autowired  
 private DAOTxBitacora daoTxBitacora;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxBitacora> lista = new ArrayList<>();  
 List<TxBitacora> objs = daoTxBitacora.findAll();  
 for(TxBitacora entity: objs) {  
 DominioTxBitacora dom = new DominioTxBitacora();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxBitacora.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxBitacora dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxBitacora> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxBitacora> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxBitacora.findByBitacoraId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxBitacora.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxBitacora.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxBitacora entity: objs) {  
 DominioTxBitacora to = new DominioTxBitacora();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxBitacora dom = (DominioTxBitacora)obj;  
 TxBitacora entity = new TxBitacora();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxBitacora.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxBitacora.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxBitacora dom = (DominioTxBitacora)obj;  
 TxBitacora entity = new TxBitacora();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxBitacora.findById(entity.getBitacoraId()).  
 orElse(new TxBitacora());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxBitacora.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxBitacora.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxBitacora dom = (DominioTxBitacora)obj;  
 TxBitacora entity = new TxBitacora();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxBitacora.findById(entity.getBitacoraId()).  
 orElse(new TxBitacora());  
 daoTxBitacora.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxBitacora.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxBitacora dom = (DominioTxBitacora)obj;  
 TxBitacora entity = new TxBitacora();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxBitacora.findById(entity.getBitacoraId()).  
 orElse(new TxBitacora());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxBitacora.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxBitacora.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxBitacora.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxBitacora existsById(Object obj) throws Exception {  
 try {  
 DominioTxBitacora dom = (DominioTxBitacora)obj;  
 TxBitacora entity = new TxBitacora();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxBitacora.existsById(entity.getBitacoraId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxBitacora.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxCincominutal.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxCincominutal;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxCincominutal;  
import mx.mbi.dgc.jpa.entity.TxCincominutal;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxCincominutal implements IfaceTxCincominutal {  
 @Autowired  
 private DAOTxCincominutal daoTxCincominutal;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxCincominutal> lista = new ArrayList<>();  
 List<TxCincominutal> objs = daoTxCincominutal.findAll();  
 for(TxCincominutal entity: objs) {  
 DominioTxCincominutal dom = new DominioTxCincominutal();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCincominutal.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxCincominutal dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxCincominutal> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxCincominutal> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxCincominutal.findByCincominutalId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxCincominutal.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxCincominutal.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxCincominutal entity: objs) {  
 DominioTxCincominutal to = new DominioTxCincominutal();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxCincominutal dom = (DominioTxCincominutal)obj;  
 TxCincominutal entity = new TxCincominutal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCincominutal.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCincominutal.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxCincominutal dom = (DominioTxCincominutal)obj;  
 TxCincominutal entity = new TxCincominutal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCincominutal.findById(entity.getCincominutalId()).  
 orElse(new TxCincominutal());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCincominutal.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCincominutal.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxCincominutal dom = (DominioTxCincominutal)obj;  
 TxCincominutal entity = new TxCincominutal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCincominutal.findById(entity.getCincominutalId()).  
 orElse(new TxCincominutal());  
 daoTxCincominutal.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCincominutal.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxCincominutal dom = (DominioTxCincominutal)obj;  
 TxCincominutal entity = new TxCincominutal();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCincominutal.findById(entity.getCincominutalId()).  
 orElse(new TxCincominutal());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCincominutal.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxCincominutal.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCincominutal.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxCincominutal existsById(Object obj) throws Exception {  
 try {  
 DominioTxCincominutal dom = (DominioTxCincominutal)obj;  
 TxCincominutal entity = new TxCincominutal();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxCincominutal.existsById(entity.getCincominutalId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCincominutal.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxCincominutalVar.java

package mx.mbi.dgc.business;  
  
import java.math.BigDecimal;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOTxCincominutalVar;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxCincominutalVar;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.jpa.entity.TxCincominutalVar;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessTxCincominutalVar implements IfaceTxCincominutalVar {  
 @Autowired  
 private DAOTxCincominutalVar daoTxCincominutalVar;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxCincominutalVar> lista = new ArrayList<>();  
 List<TxCincominutalVar> objs = daoTxCincominutalVar.findAll();  
 for(TxCincominutalVar entity: objs) {  
 DominioTxCincominutalVar dom = new DominioTxCincominutalVar();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCincominutalVar.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public List<DominioTxCincominutalVar> findByEstatusCnNotAndFcPrimera(BigDecimal estatusCn,CatPlanta catPlanta) throws Exception {  
 try {  
 List<DominioTxCincominutalVar> lista = new ArrayList<>();  
 System.out.println("Obteniendo cincominutales de planta: "+catPlanta.getPlantaId());  
 List<TxCincominutalVar> objs = daoTxCincominutalVar.findByEstatusCnNotAndFcPrimeraIsNullAndTxCincominutal\_CatPlanta(estatusCn,catPlanta);  
 System.out.println("Cincominutales obtenidos :"+objs.size());  
 for(TxCincominutalVar entity: objs) {  
 DominioTxCincominutalVar dom = new DominioTxCincominutalVar();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCincominutalVar.findAll: "+ex);  
 }  
 }  
 @Override  
 public DominioRespuesta allPagination(DominioTxCincominutalVar dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxCincominutalVar> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxCincominutalVar> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxCincominutalVar.findByCincominutalVarId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxCincominutalVar.findByValorOriginalContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxCincominutalVar.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxCincominutalVar entity: objs) {  
 DominioTxCincominutalVar to = new DominioTxCincominutalVar();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxCincominutalVar dom = (DominioTxCincominutalVar)obj;  
 TxCincominutalVar entity = new TxCincominutalVar();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCincominutalVar.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCincominutalVar.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxCincominutalVar dom = (DominioTxCincominutalVar)obj;  
 TxCincominutalVar entity = new TxCincominutalVar();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCincominutalVar.findById(entity.getCincominutalVarId()).  
 orElse(new TxCincominutalVar());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCincominutalVar.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCincominutalVar.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxCincominutalVar dom = (DominioTxCincominutalVar)obj;  
 TxCincominutalVar entity = new TxCincominutalVar();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCincominutalVar.findById(entity.getCincominutalVarId()).  
 orElse(new TxCincominutalVar());  
 daoTxCincominutalVar.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCincominutalVar.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxCincominutalVar dom = (DominioTxCincominutalVar)obj;  
 TxCincominutalVar entity = new TxCincominutalVar();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCincominutalVar.findById(entity.getCincominutalVarId()).  
 orElse(new TxCincominutalVar());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCincominutalVar.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxCincominutalVar.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCincominutalVar.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxCincominutalVar existsById(Object obj) throws Exception {  
 try {  
 DominioTxCincominutalVar dom = (DominioTxCincominutalVar)obj;  
 TxCincominutalVar entity = new TxCincominutalVar();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxCincominutalVar.existsById(entity.getCincominutalVarId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCincominutalVar.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxCrudos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxCrudos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxCrudos;  
import mx.mbi.dgc.jpa.entity.TxCrudos;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxCrudos implements IfaceTxCrudos {  
 @Autowired  
 private DAOTxCrudos daoTxCrudos;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxCrudos> lista = new ArrayList<>();  
 List<TxCrudos> objs = daoTxCrudos.findAll();  
 for(TxCrudos entity: objs) {  
 DominioTxCrudos dom = new DominioTxCrudos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudos.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxCrudos dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxCrudos> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxCrudos> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxCrudos.findByDia(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxCrudos.findByDiaContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxCrudos.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxCrudos entity: objs) {  
 DominioTxCrudos to = new DominioTxCrudos();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxCrudos dom = (DominioTxCrudos)obj;  
 TxCrudos entity = new TxCrudos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCrudos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudos.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxCrudos dom = (DominioTxCrudos)obj;  
 TxCrudos entity = new TxCrudos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCrudos.findById(entity.getCrudosId()).  
 orElse(new TxCrudos());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCrudos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudos.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxCrudos dom = (DominioTxCrudos)obj;  
 TxCrudos entity = new TxCrudos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCrudos.findById(entity.getCrudosId()).  
 orElse(new TxCrudos());  
 daoTxCrudos.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudos.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxCrudos dom = (DominioTxCrudos)obj;  
 TxCrudos entity = new TxCrudos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCrudos.findById(entity.getCrudosId()).  
 orElse(new TxCrudos());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxCrudos.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudos.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxCrudos existsById(Object obj) throws Exception {  
 try {  
 DominioTxCrudos dom = (DominioTxCrudos)obj;  
 TxCrudos entity = new TxCrudos();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxCrudos.existsById(entity.getCrudosId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxCrudosLimpios.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxCrudosLimpios;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxCrudosLimpios;  
import mx.mbi.dgc.jpa.entity.TxCrudosLimpios;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxCrudosLimpios implements IfaceTxCrudosLimpios {  
 @Autowired  
 private DAOTxCrudosLimpios daoTxCrudosLimpios;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxCrudosLimpios> lista = new ArrayList<>();  
 List<TxCrudosLimpios> objs = daoTxCrudosLimpios.findAll();  
 for(TxCrudosLimpios entity: objs) {  
 DominioTxCrudosLimpios dom = new DominioTxCrudosLimpios();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudosLimpios.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxCrudosLimpios dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxCrudosLimpios> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxCrudosLimpios> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxCrudosLimpios.findByDia(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxCrudosLimpios.findByMinutoContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxCrudosLimpios.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxCrudosLimpios entity: objs) {  
 DominioTxCrudosLimpios to = new DominioTxCrudosLimpios();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxCrudosLimpios dom = (DominioTxCrudosLimpios)obj;  
 TxCrudosLimpios entity = new TxCrudosLimpios();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCrudosLimpios.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudosLimpios.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxCrudosLimpios dom = (DominioTxCrudosLimpios)obj;  
 TxCrudosLimpios entity = new TxCrudosLimpios();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCrudosLimpios.findById(entity.getFirma()).  
 orElse(new TxCrudosLimpios());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCrudosLimpios.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudosLimpios.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxCrudosLimpios dom = (DominioTxCrudosLimpios)obj;  
 TxCrudosLimpios entity = new TxCrudosLimpios();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCrudosLimpios.findById(entity.getFirma()).  
 orElse(new TxCrudosLimpios());  
 daoTxCrudosLimpios.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudosLimpios.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxCrudosLimpios dom = (DominioTxCrudosLimpios)obj;  
 TxCrudosLimpios entity = new TxCrudosLimpios();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCrudosLimpios.findById(entity.getFirma()).  
 orElse(new TxCrudosLimpios());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudosLimpios.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxCrudosLimpios.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudosLimpios.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxCrudosLimpios existsById(Object obj) throws Exception {  
 try {  
 DominioTxCrudosLimpios dom = (DominioTxCrudosLimpios)obj;  
 TxCrudosLimpios entity = new TxCrudosLimpios();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxCrudosLimpios.existsById(entity.getFirma()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudosLimpios.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxCrudosSol.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxCrudosSol;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxCrudosSol;  
import mx.mbi.dgc.jpa.entity.TxCrudosSol;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxCrudosSol implements IfaceTxCrudosSol {  
 @Autowired  
 private DAOTxCrudosSol daoTxCrudosSol;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxCrudosSol> lista = new ArrayList<>();  
 List<TxCrudosSol> objs = daoTxCrudosSol.findAll();  
 for(TxCrudosSol entity: objs) {  
 DominioTxCrudosSol dom = new DominioTxCrudosSol();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudosSol.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxCrudosSol dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxCrudosSol> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxCrudosSol> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxCrudosSol.findByDia(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxCrudosSol.findByDiaContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxCrudosSol.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxCrudosSol entity: objs) {  
 DominioTxCrudosSol to = new DominioTxCrudosSol();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxCrudosSol dom = (DominioTxCrudosSol)obj;  
 TxCrudosSol entity = new TxCrudosSol();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCrudosSol.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudosSol.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxCrudosSol dom = (DominioTxCrudosSol)obj;  
 TxCrudosSol entity = new TxCrudosSol();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCrudosSol.findById(entity.getCrudosId()).  
 orElse(new TxCrudosSol());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCrudosSol.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudosSol.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxCrudosSol dom = (DominioTxCrudosSol)obj;  
 TxCrudosSol entity = new TxCrudosSol();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCrudosSol.findById(entity.getCrudosId()).  
 orElse(new TxCrudosSol());  
 daoTxCrudosSol.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudosSol.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxCrudosSol dom = (DominioTxCrudosSol)obj;  
 TxCrudosSol entity = new TxCrudosSol();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxCrudosSol.findById(entity.getCrudosId()).  
 orElse(new TxCrudosSol());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudosSol.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxCrudosSol.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudosSol.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxCrudosSol existsById(Object obj) throws Exception {  
 try {  
 DominioTxCrudosSol dom = (DominioTxCrudosSol)obj;  
 TxCrudosSol entity = new TxCrudosSol();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxCrudosSol.existsById(entity.getCrudosId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxCrudosSol.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxDocumentos.java

package mx.mbi.dgc.business;  
  
import java.text.DateFormat;  
import java.text.SimpleDateFormat;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.HashSet;  
import java.util.List;  
import java.util.Set;  
  
import mx.mbi.dgc.business.facturacion.BusinessFacturacion;  
import mx.mbi.dgc.dao.DAOCatFacturaProductos;  
import mx.mbi.dgc.dao.DAOTxDocumentos;  
import mx.mbi.dgc.dominio.DominioCatFacturaProductos;  
import mx.mbi.dgc.dominio.DominioCatServicioImpuestos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxDocumentos;  
import mx.mbi.dgc.jpa.entity.CatFacturaProductos;  
import mx.mbi.dgc.jpa.entity.TxDocumentos;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessTxDocumentos implements IfaceTxDocumentos {  
 @Autowired  
 private DAOTxDocumentos daoTxDocumentos;  
  
 @Autowired  
 private DAOCatFacturaProductos daoCatFacturaProductos;  
 @Autowired  
 private BusinessFacturacion businessFacturacion;  
  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxDocumentos> lista = new ArrayList<>();  
 List<TxDocumentos> objs = daoTxDocumentos.findAll();  
 for (TxDocumentos entity : objs) {  
 DominioTxDocumentos dom = new DominioTxDocumentos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxDocumentos.findAll: " + ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxDocumentos dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxDocumentos> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxDocumentos> objs = null;  
 if (dom.getSearch() != null && dom.getSearch().getValue() != null &&  
 ("" + dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs =  
 daoTxDocumentos.findByDocumentoId(new java.math.BigDecimal("" + dom.getSearch().getValue()),  
 PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC : Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex) {  
 try {  
 objs =  
 daoTxDocumentos.findBySerieContaining("" + dom.getSearch().getValue(),  
 PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC :  
 Sort.Direction.DESC,  
 dom.getCampos()  
 .get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex2) {  
 }  
 }  
 } else {  
 objs =  
 daoTxDocumentos.findAll(PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC : Sort.Direction.DESC,  
 dom.getCampos()  
.get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 }  
 for (TxDocumentos entity : objs) {  
 DominioTxDocumentos to = new DominioTxDocumentos();  
 JRUtil.copyProperties(to, entity);  
 DateFormat df = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");  
 String dateStr = df.format(to.getFechaHoraCreacion());  
 String dateCStr = "";  
 if (to.getFechaHoraCancelacion() != null) {  
 dateCStr = df.format(to.getFechaHoraCancelacion());  
 }  
 to = calculaCuenta(to);  
 to.setFechaCreacionStr(dateStr);  
 to.setFechaCancelacionStr(dateCStr);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: " + ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxDocumentos dom = (DominioTxDocumentos) obj;  
 TxDocumentos entity = new TxDocumentos();  
 JRUtil.copyProperties(entity, dom);  
 List<DominioCatFacturaProductos> lista = dom.getCatFacturaProductoses();  
 Set<CatFacturaProductos> entidades = new HashSet<CatFacturaProductos>();  
  
  
 entity.setCatFacturaProductoses(entidades);  
 entity = daoTxDocumentos.save(entity);  
  
 for (DominioCatFacturaProductos dom2 : lista) {  
 CatFacturaProductos entity2 = new CatFacturaProductos();  
 JRUtil.copyProperties(entity2, dom2);  
 entity2.getTxDocumentos().setDocumentoId(entity.getDocumentoId());  
 //entidades.add(entity2);  
 System.out.println("descuento:" + entity2.getDescuento());  
 System.out.println("documento en producto:" + entity2.getTxDocumentos().getFolio());  
 System.out.println("documento id:" + entity2.getTxDocumentos().getDocumentoId());  
 daoCatFacturaProductos.save(entity2);  
 }  
  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxDocumentos.save: " + ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxDocumentos dom = (DominioTxDocumentos) obj;  
 TxDocumentos entity = new TxDocumentos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxDocumentos.findById(entity.getDocumentoId()).orElse(new TxDocumentos());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxDocumentos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxDocumentos.update: " + ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxDocumentos dom = (DominioTxDocumentos) obj;  
 TxDocumentos entity = new TxDocumentos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxDocumentos.findById(entity.getDocumentoId()).orElse(new TxDocumentos());  
 daoTxDocumentos.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxDocumentos.delete: " + ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxDocumentos dom = (DominioTxDocumentos) obj;  
 TxDocumentos entity = new TxDocumentos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxDocumentos.findById(entity.getDocumentoId()).orElse(new TxDocumentos());  
 // Set<CatFacturaProductos> productos = daoCatFacturaProductos.findByTxDocumentos(entity);  
 // System.out.println("obteniendo productos de documento id : " + entity.getDocumentoId());  
 //System.out.println("tamaño de productos en entity: " + productos.size());  
 //entity.setCatFacturaProductoses(productos);  
 JRUtil.copyProperties(dom, entity);  
 //System.out.println("tamaño de productos en dom: " + dom.getCatFacturaProductoses().size());  
 //Obteniendo Servicios para el TxDocumento consultado  
 List<DominioCatFacturaProductos> productos = businessFacturacion.findFacturaProductosByTxDocumento(dom);  
  
  
 //Obteniendo Impuestos por cada servicio  
 List<DominioCatFacturaProductos> listaFinal = new ArrayList<>();  
 for (DominioCatFacturaProductos domProducto : productos) {  
 List<DominioCatServicioImpuestos> impuestos =  
 businessFacturacion.findImpuestosByServicio(domProducto.getCatServicios());  
 domProducto.getCatServicios().setCatServicioImpuestoses(impuestos);  
  
 System.out.println("tamaño de impuestos en dom: " + domProducto.getCatServicios()  
 .getCatServicioImpuestoses()  
 .size());  
 listaFinal.add(domProducto);  
  
 }  
  
 dom.setCatFacturaProductoses(listaFinal);  
 System.out.println("tamaño de productos en dom: " + dom.getCatFacturaProductoses().size());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxDocumentos.findById: " + ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxDocumentos.count();  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxDocumentos.count: " + ex);  
 }  
 }  
  
 @Override  
 public DominioTxDocumentos existsById(Object obj) throws Exception {  
 try {  
 DominioTxDocumentos dom = (DominioTxDocumentos) obj;  
 TxDocumentos entity = new TxDocumentos();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxDocumentos.existsById(entity.getDocumentoId()));  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxDocumentos.findById: " + ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
  
 private DominioTxDocumentos calculaCuenta(DominioTxDocumentos dom) throws Exception {  
 List<DominioCatFacturaProductos> productos = businessFacturacion.findFacturaProductosByTxDocumento(dom);  
 double subtotal = 0;  
 double impuestoT = 0;  
 double total = 0;  
 double iva = 0;  
 for (DominioCatFacturaProductos domProducto : productos) {  
 List<DominioCatServicioImpuestos> impuestos =  
 businessFacturacion.findImpuestosByServicio(domProducto.getCatServicios());  
 for (DominioCatServicioImpuestos impuesto : impuestos) {  
 if (impuesto.getCatImpuestos()  
 .getNombre()  
 .contains("IVA")) {  
 impuestoT += impuesto.getCatImpuestos()  
 .getPorcentaje()  
 .doubleValue();  
 }  
 }  
  
 subtotal += domProducto.getImporte().doubleValue();  
 iva = subtotal \* impuestoT;  
 total = subtotal \* (impuestoT + 1);  
 }  
 dom.setSubtotal(String.valueOf(subtotal));  
 dom.setIva(String.valueOf(iva));  
 dom.setTotal(String.valueOf(total));  
 return dom;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxErrorCargaCrudos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxErrorCargaCrudos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxErrorCargaCrudos;  
import mx.mbi.dgc.jpa.entity.TxErrorCargaCrudos;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxErrorCargaCrudos implements IfaceTxErrorCargaCrudos {  
 @Autowired  
 private DAOTxErrorCargaCrudos daoTxErrorCargaCrudos;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxErrorCargaCrudos> lista = new ArrayList<>();  
 List<TxErrorCargaCrudos> objs = daoTxErrorCargaCrudos.findAll();  
 for(TxErrorCargaCrudos entity: objs) {  
 DominioTxErrorCargaCrudos dom = new DominioTxErrorCargaCrudos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxErrorCargaCrudos.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxErrorCargaCrudos dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxErrorCargaCrudos> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxErrorCargaCrudos> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxErrorCargaCrudos.findByTipoError(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxErrorCargaCrudos.findByErrmContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxErrorCargaCrudos.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxErrorCargaCrudos entity: objs) {  
 DominioTxErrorCargaCrudos to = new DominioTxErrorCargaCrudos();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxErrorCargaCrudos dom = (DominioTxErrorCargaCrudos)obj;  
 TxErrorCargaCrudos entity = new TxErrorCargaCrudos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxErrorCargaCrudos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxErrorCargaCrudos.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxErrorCargaCrudos dom = (DominioTxErrorCargaCrudos)obj;  
 TxErrorCargaCrudos entity = new TxErrorCargaCrudos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxErrorCargaCrudos.findById(entity.getErrorCargaCrudosId()).  
 orElse(new TxErrorCargaCrudos());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxErrorCargaCrudos.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxErrorCargaCrudos.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxErrorCargaCrudos dom = (DominioTxErrorCargaCrudos)obj;  
 TxErrorCargaCrudos entity = new TxErrorCargaCrudos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxErrorCargaCrudos.findById(entity.getErrorCargaCrudosId()).  
 orElse(new TxErrorCargaCrudos());  
 daoTxErrorCargaCrudos.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxErrorCargaCrudos.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxErrorCargaCrudos dom = (DominioTxErrorCargaCrudos)obj;  
 TxErrorCargaCrudos entity = new TxErrorCargaCrudos();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxErrorCargaCrudos.findById(entity.getErrorCargaCrudosId()).  
 orElse(new TxErrorCargaCrudos());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxErrorCargaCrudos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxErrorCargaCrudos.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxErrorCargaCrudos.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxErrorCargaCrudos existsById(Object obj) throws Exception {  
 try {  
 DominioTxErrorCargaCrudos dom = (DominioTxErrorCargaCrudos)obj;  
 TxErrorCargaCrudos entity = new TxErrorCargaCrudos();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxErrorCargaCrudos.existsById(entity.getErrorCargaCrudosId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxErrorCargaCrudos.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxErrorFormato.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxErrorFormato;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxErrorFormato;  
import mx.mbi.dgc.jpa.entity.TxErrorFormato;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxErrorFormato implements IfaceTxErrorFormato {  
 @Autowired  
 private DAOTxErrorFormato daoTxErrorFormato;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxErrorFormato> lista = new ArrayList<>();  
 List<TxErrorFormato> objs = daoTxErrorFormato.findAll();  
 for(TxErrorFormato entity: objs) {  
 DominioTxErrorFormato dom = new DominioTxErrorFormato();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxErrorFormato.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxErrorFormato dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxErrorFormato> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxErrorFormato> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxErrorFormato.findByErrorFormatoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxErrorFormato.findByValorContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxErrorFormato.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxErrorFormato entity: objs) {  
 DominioTxErrorFormato to = new DominioTxErrorFormato();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxErrorFormato dom = (DominioTxErrorFormato)obj;  
 TxErrorFormato entity = new TxErrorFormato();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxErrorFormato.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxErrorFormato.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxErrorFormato dom = (DominioTxErrorFormato)obj;  
 TxErrorFormato entity = new TxErrorFormato();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxErrorFormato.findById(entity.getErrorFormatoId()).  
 orElse(new TxErrorFormato());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxErrorFormato.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxErrorFormato.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxErrorFormato dom = (DominioTxErrorFormato)obj;  
 TxErrorFormato entity = new TxErrorFormato();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxErrorFormato.findById(entity.getErrorFormatoId()).  
 orElse(new TxErrorFormato());  
 daoTxErrorFormato.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxErrorFormato.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxErrorFormato dom = (DominioTxErrorFormato)obj;  
 TxErrorFormato entity = new TxErrorFormato();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxErrorFormato.findById(entity.getErrorFormatoId()).  
 orElse(new TxErrorFormato());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxErrorFormato.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxErrorFormato.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxErrorFormato.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxErrorFormato existsById(Object obj) throws Exception {  
 try {  
 DominioTxErrorFormato dom = (DominioTxErrorFormato)obj;  
 TxErrorFormato entity = new TxErrorFormato();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxErrorFormato.existsById(entity.getErrorFormatoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxErrorFormato.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxEscenario.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxEscenario;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxEscenario;  
import mx.mbi.dgc.jpa.entity.TxEscenario;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxEscenario implements IfaceTxEscenario {  
 @Autowired  
 private DAOTxEscenario daoTxEscenario;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxEscenario> lista = new ArrayList<>();  
 List<TxEscenario> objs = daoTxEscenario.findAll();  
 for(TxEscenario entity: objs) {  
 DominioTxEscenario dom = new DominioTxEscenario();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenario.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxEscenario dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxEscenario> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxEscenario> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxEscenario.findByEscenarioId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxEscenario.findByNombreContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxEscenario.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxEscenario entity: objs) {  
 DominioTxEscenario to = new DominioTxEscenario();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxEscenario dom = (DominioTxEscenario)obj;  
 TxEscenario entity = new TxEscenario();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEscenario.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenario.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxEscenario dom = (DominioTxEscenario)obj;  
 TxEscenario entity = new TxEscenario();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEscenario.findById(entity.getEscenarioId()).  
 orElse(new TxEscenario());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEscenario.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenario.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxEscenario dom = (DominioTxEscenario)obj;  
 TxEscenario entity = new TxEscenario();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEscenario.findById(entity.getEscenarioId()).  
 orElse(new TxEscenario());  
 daoTxEscenario.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenario.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxEscenario dom = (DominioTxEscenario)obj;  
 TxEscenario entity = new TxEscenario();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEscenario.findById(entity.getEscenarioId()).  
 orElse(new TxEscenario());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenario.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxEscenario.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenario.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxEscenario existsById(Object obj) throws Exception {  
 try {  
 DominioTxEscenario dom = (DominioTxEscenario)obj;  
 TxEscenario entity = new TxEscenario();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxEscenario.existsById(entity.getEscenarioId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenario.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxEscenariosBitacora.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxEscenariosBitacora;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxEscenariosBitacora;  
import mx.mbi.dgc.jpa.entity.TxEscenariosBitacora;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxEscenariosBitacora implements IfaceTxEscenariosBitacora {  
 @Autowired  
 private DAOTxEscenariosBitacora daoTxEscenariosBitacora;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxEscenariosBitacora> lista = new ArrayList<>();  
 List<TxEscenariosBitacora> objs = daoTxEscenariosBitacora.findAll();  
 for(TxEscenariosBitacora entity: objs) {  
 DominioTxEscenariosBitacora dom = new DominioTxEscenariosBitacora();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenariosBitacora.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxEscenariosBitacora dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxEscenariosBitacora> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxEscenariosBitacora> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxEscenariosBitacora.findByEscenarioBitacoraId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxEscenariosBitacora.findByEscenarioContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxEscenariosBitacora.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxEscenariosBitacora entity: objs) {  
 DominioTxEscenariosBitacora to = new DominioTxEscenariosBitacora();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxEscenariosBitacora dom = (DominioTxEscenariosBitacora)obj;  
 TxEscenariosBitacora entity = new TxEscenariosBitacora();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEscenariosBitacora.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenariosBitacora.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxEscenariosBitacora dom = (DominioTxEscenariosBitacora)obj;  
 TxEscenariosBitacora entity = new TxEscenariosBitacora();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEscenariosBitacora.findById(entity.getEscenarioBitacoraId()).  
 orElse(new TxEscenariosBitacora());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEscenariosBitacora.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenariosBitacora.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxEscenariosBitacora dom = (DominioTxEscenariosBitacora)obj;  
 TxEscenariosBitacora entity = new TxEscenariosBitacora();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEscenariosBitacora.findById(entity.getEscenarioBitacoraId()).  
 orElse(new TxEscenariosBitacora());  
 daoTxEscenariosBitacora.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenariosBitacora.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxEscenariosBitacora dom = (DominioTxEscenariosBitacora)obj;  
 TxEscenariosBitacora entity = new TxEscenariosBitacora();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEscenariosBitacora.findById(entity.getEscenarioBitacoraId()).  
 orElse(new TxEscenariosBitacora());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenariosBitacora.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxEscenariosBitacora.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenariosBitacora.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxEscenariosBitacora existsById(Object obj) throws Exception {  
 try {  
 DominioTxEscenariosBitacora dom = (DominioTxEscenariosBitacora)obj;  
 TxEscenariosBitacora entity = new TxEscenariosBitacora();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxEscenariosBitacora.existsById(entity.getEscenarioBitacoraId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenariosBitacora.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxEscenarioVariable.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxEscenarioVariable;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxEscenarioVariable;  
import mx.mbi.dgc.jpa.entity.TxEscenarioVariable;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxEscenarioVariable implements IfaceTxEscenarioVariable {  
 @Autowired  
 private DAOTxEscenarioVariable daoTxEscenarioVariable;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxEscenarioVariable> lista = new ArrayList<>();  
 List<TxEscenarioVariable> objs = daoTxEscenarioVariable.findAll();  
 for(TxEscenarioVariable entity: objs) {  
 DominioTxEscenarioVariable dom = new DominioTxEscenarioVariable();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenarioVariable.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxEscenarioVariable dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxEscenarioVariable> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxEscenarioVariable> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxEscenarioVariable.findByEscenarioVariableId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxEscenarioVariable.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxEscenarioVariable.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxEscenarioVariable entity: objs) {  
 DominioTxEscenarioVariable to = new DominioTxEscenarioVariable();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxEscenarioVariable dom = (DominioTxEscenarioVariable)obj;  
 TxEscenarioVariable entity = new TxEscenarioVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEscenarioVariable.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenarioVariable.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxEscenarioVariable dom = (DominioTxEscenarioVariable)obj;  
 TxEscenarioVariable entity = new TxEscenarioVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEscenarioVariable.findById(entity.getEscenarioVariableId()).  
 orElse(new TxEscenarioVariable());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEscenarioVariable.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenarioVariable.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxEscenarioVariable dom = (DominioTxEscenarioVariable)obj;  
 TxEscenarioVariable entity = new TxEscenarioVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEscenarioVariable.findById(entity.getEscenarioVariableId()).  
 orElse(new TxEscenarioVariable());  
 daoTxEscenarioVariable.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenarioVariable.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxEscenarioVariable dom = (DominioTxEscenarioVariable)obj;  
 TxEscenarioVariable entity = new TxEscenarioVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEscenarioVariable.findById(entity.getEscenarioVariableId()).  
 orElse(new TxEscenarioVariable());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenarioVariable.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxEscenarioVariable.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenarioVariable.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxEscenarioVariable existsById(Object obj) throws Exception {  
 try {  
 DominioTxEscenarioVariable dom = (DominioTxEscenarioVariable)obj;  
 TxEscenarioVariable entity = new TxEscenarioVariable();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxEscenarioVariable.existsById(entity.getEscenarioVariableId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxEscenarioVariable.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxEvento.java

package mx.mbi.dgc.business;  
  
import java.math.BigDecimal;  
  
import java.text.DateFormat;  
import java.text.SimpleDateFormat;  
  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.Collections;  
import java.util.Date;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOTxAdjuntos;  
import mx.mbi.dgc.dao.DAOTxBitacora;  
import mx.mbi.dgc.dao.DAOTxEvento;  
import mx.mbi.dgc.dao.DAOTxEventoDetalles;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxEvento;  
import mx.mbi.dgc.dominio.negocio.bitacora.DominioBusquedaEventos;  
import mx.mbi.dgc.dominio.negocio.bitacora.DominioEventoBitacora;  
import mx.mbi.dgc.jpa.entity.CatConfiguracionOpciones;  
import mx.mbi.dgc.jpa.entity.TxAdjuntos;  
import mx.mbi.dgc.jpa.entity.TxBitacora;  
import mx.mbi.dgc.jpa.entity.TxEvento;  
import mx.mbi.dgc.jpa.entity.TxEventoDetalles;  
import mx.mbi.dgc.jpa.entity.TxUsuario;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessTxEvento implements IfaceTxEvento {  
 @Autowired  
 private DAOTxEvento daoTxEvento;  
  
 @Autowired  
 private DAOTxAdjuntos daoTxAdjuntos;  
 @Autowired  
 private DAOTxBitacora daoTxBitacora;  
  
 @Autowired  
 private DAOTxEventoDetalles daoTxEventoDetalles;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxEvento> lista = new ArrayList<>();  
 List<TxEvento> objs = daoTxEvento.findAll();  
 for (TxEvento entity : objs) {  
 DominioTxEvento dom = new DominioTxEvento();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxEvento.findAll: " + ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxEvento dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxEvento> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxEvento> objs = null;  
 if (dom.getSearch() != null && dom.getSearch().getValue() != null &&  
 ("" + dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs =  
 daoTxEvento.findByEventoId(new java.math.BigDecimal("" + dom.getSearch().getValue()),  
 PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC : Sort.Direction.DESC,  
 dom.getCampos().get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex) {  
 try {  
 objs =  
 daoTxEvento.findByInformesContaining("" + dom.getSearch().getValue(),  
 PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC :  
 Sort.Direction.DESC,  
 dom.getCampos()  
 .get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 } catch (Exception ex2) {  
 }  
 }  
 } else {  
 objs =  
 daoTxEvento.findAll(PageRequest.of(page, dom.getLength(),  
 dom.getOrder()  
 .get(0)  
 .getDir()  
 .equals("asc") ?  
 Sort.Direction.ASC : Sort.Direction.DESC,  
 dom.getCampos()  
 .get(dom.getOrder()  
 .get(0)  
 .getColumn())));  
 }  
 for (TxEvento entity : objs) {  
 DominioTxEvento to = new DominioTxEvento();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: " + ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxEvento dom = (DominioTxEvento) obj;  
 TxEvento entity = new TxEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEvento.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxEvento.save: " + ex);  
 }  
 }  
  
 @Override  
 public DominioTxEvento guardarEvento(DominioEventoBitacora dom) throws Exception {  
 try {  
 DominioTxEvento res = new DominioTxEvento();  
 TxEvento entity = new TxEvento();  
 JRUtil.copyProperties(entity, dom.getEvento());  
  
  
 Calendar cal = Calendar.getInstance();  
 cal.setTime(dom.getEvento().getFechaInicio());  
  
 cal.set(Calendar.DAY\_OF\_MONTH, 1);  
 Date fecha = cal.getTime();  
  
 DateFormat outputFormatter = new SimpleDateFormat("MM/dd/yyyy");  
 String output = outputFormatter.format(fecha);  
  
 DateFormat inputFormatter = new SimpleDateFormat("MM/dd/yyyy");  
 Date date = inputFormatter.parse(output);  
 // System.out.println(fecha);  
 // System.out.println(date);  
 // System.out.println(dom.getEvento().getFechaInicio());  
 TxBitacora bitacora = daoTxBitacora.findByFecha(date);  
 // System.out.println(bitacora);  
 // Set<TxAdjuntos> setArchivos = new HashSet<TxAdjuntos>(archivos);  
 // entity.setTxAdjuntosSet(setArchivos);  
 // System.out.println(entity.getFechaInicio());  
 // System.out.println(redondeaFecha(entity.getFechaInicio()));  
 if (entity.getFechaInicio() != null) {  
 Date fechaInicio = new Date(entity.getFechaInicio().getTime());  
 entity.setFechaInicioRedondeo(redondeaFecha(fechaInicio));  
 }  
 if (entity.getFechaFinal() != null) {  
 Date fechaFinal = new Date(entity.getFechaFinal().getTime());  
 entity.setFechaFinalRedondeo(redondeaFecha(fechaFinal));  
 }  
 entity.setBitacoraId(bitacora);  
 entity = daoTxEvento.save(entity);  
  
 //creando Opciones detalles a guardar  
 // List<TxEventoDetalles> eventoDetallesLista = new ArrayList<>();  
 for (TxEventoDetalles eventoDetalle : daoTxEventoDetalles.findByEventoId(entity)) {  
 daoTxEventoDetalles.delete(eventoDetalle);  
 }  
 for (BigDecimal opcionDetalleId : dom.getDetalles()) {  
 // System.out.println("opcionDetalleId: " + opcionDetalleId);  
 TxEventoDetalles eventoDetalle = new TxEventoDetalles();  
 CatConfiguracionOpciones configOpciones = new CatConfiguracionOpciones();  
 configOpciones.setConfiguracionOpcionesId(opcionDetalleId);  
 eventoDetalle.setConfiguracionOpcionesId(configOpciones);  
 eventoDetalle.setEventoId(entity);  
 daoTxEventoDetalles.save(eventoDetalle);  
 // eventoDetallesLista.add(eventoDetalle);  
 }  
  
  
 // Set<TxEventoDetalles> setEventoDetalles = new HashSet<TxEventoDetalles>(eventoDetallesLista);  
 // entity.setTxEventoDetallesSet(setEventoDetalles);  
  
  
 //Creando archivos y guardando en entity  
 //List<TxAdjuntos> archivos = new ArrayList<>();  
 for (String docPath : dom.getDocumentos()) {  
 TxAdjuntos archivo = new TxAdjuntos();  
 archivo.setReferencia(docPath);  
 archivo.setEventoId(entity);  
 //archivos.add(archivo);  
 daoTxAdjuntos.save(archivo);  
 }  
  
  
 JRUtil.copyProperties(res, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return res;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 return null;  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxEvento dom = (DominioTxEvento) obj;  
 TxEvento entity = new TxEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEvento.findById(entity.getEventoId()).orElse(new TxEvento());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEvento.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxEvento.update: " + ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxEvento dom = (DominioTxEvento) obj;  
 TxEvento entity = new TxEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEvento.findById(entity.getEventoId()).orElse(new TxEvento());  
 daoTxEvento.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxEvento.delete: " + ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxEvento dom = (DominioTxEvento) obj;  
 TxEvento entity = new TxEvento();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEvento.findById(entity.getEventoId()).orElse(new TxEvento());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 return null;  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxEvento.count();  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxEvento.count: " + ex);  
 }  
 }  
  
 @Override  
 public DominioTxEvento existsById(Object obj) throws Exception {  
 try {  
 DominioTxEvento dom = (DominioTxEvento) obj;  
 TxEvento entity = new TxEvento();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxEvento.existsById(entity.getEventoId()));  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxEvento.findById: " + ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
  
 @Override  
 public List<DominioTxEvento> findByBitacora(DominioBusquedaEventos dom) throws Exception {  
 try {  
 System.out.println("Buscando por fecha: " + dom.getFecha());  
 TxBitacora bitacora = daoTxBitacora.findByFecha(dom.getFecha());  
 TxUsuario usuario = new TxUsuario();  
 JRUtil.copyProperties(usuario, dom.getUsuario());  
 if (bitacora != null)  
 System.out.println(bitacora.getBitacoraId());  
 else {  
 System.out.println("nulo bitacora");  
 }  
 List<TxEvento> objs = daoTxEvento.findByBitacoraIdAndUsuarioId(bitacora, usuario);  
 List<DominioTxEvento> lista = new ArrayList<>();  
 for (TxEvento entity : objs) {  
 DominioTxEvento dom2 = new DominioTxEvento();  
 JRUtil.copyProperties(dom2, entity);  
 lista.add(dom2);  
 }  
 return lista;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 return null;  
 }  
 }  
  
 private Date redondeaFecha(Date date) {  
 int minutosMod = date.getMinutes() % 10;  
 int milisegundos = 0;  
 if (minutosMod < 2) {  
 milisegundos = (minutosMod \* 60 \* 1000) + (date.getSeconds() \* 1000);  
 date.setTime(date.getTime() - milisegundos);  
 }  
 if (minutosMod > 2) {  
 if (minutosMod > 5) {  
 if (minutosMod < 7) {  
 milisegundos = ((minutosMod \* 60 \* 1000) - 300000) + (date.getSeconds() \* 1000);  
 date.setTime(date.getTime() - milisegundos);  
 }  
 if (minutosMod > 7) {  
 milisegundos = 600000 - ((minutosMod \* 60 \* 1000) + (date.getSeconds() \* 1000));  
 date.setTime(date.getTime() + milisegundos);  
 }  
 if (minutosMod == 7) {  
 if (date.getSeconds() < 30) {  
 milisegundos = ((minutosMod \* 60 \* 1000) - 300000) + (date.getSeconds() \* 1000);  
 date.setTime(date.getTime() - milisegundos);  
 } else {  
 milisegundos = 600000 - ((minutosMod \* 60 \* 1000) + (date.getSeconds() \* 1000));  
 date.setTime(date.getTime() + milisegundos);  
 }  
 }  
 } else {  
 milisegundos = 300000 - ((minutosMod \* 60 \* 1000) + (date.getSeconds() \* 1000));  
 date.setTime(date.getTime() + milisegundos);  
 }  
 }  
 if (minutosMod == 2) {  
 if (date.getSeconds() < 30) {  
  
 milisegundos = (minutosMod \* 60 \* 1000) + (date.getSeconds() \* 1000);  
 date.setTime(date.getTime() - milisegundos);  
 } else {  
 milisegundos = 300000 - ((minutosMod \* 60 \* 1000) + (date.getSeconds() \* 1000));  
 date.setTime(date.getTime() + milisegundos);  
 }  
 }  
  
 return date;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxEventoDetalles.java

package mx.mbi.dgc.business;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOTxEventoDetalles;  
import mx.mbi.dgc.dominio.DominioTxEvento;  
import mx.mbi.dgc.dominio.DominioTxEventoDetalles;  
import mx.mbi.dgc.jpa.entity.TxEvento;  
import mx.mbi.dgc.jpa.entity.TxEventoDetalles;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessTxEventoDetalles implements IfaceTxEventoDetalles {  
 @Autowired  
 private DAOTxEventoDetalles daoTxEventoDetalles;  
  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxEventoDetalles> lista = new ArrayList<>();  
 List<TxEventoDetalles> objs = daoTxEventoDetalles.findAll();  
 for (TxEventoDetalles entity : objs) {  
 DominioTxEventoDetalles dom = new DominioTxEventoDetalles();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxEventoDetalles.findAll: " + ex);  
 }  
 }  
  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxEventoDetalles dom = (DominioTxEventoDetalles) obj;  
 TxEventoDetalles entity = new TxEventoDetalles();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEventoDetalles.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxEventoDetalles.save: " + ex);  
 }  
 }  
  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxEventoDetalles dom = (DominioTxEventoDetalles) obj;  
 TxEventoDetalles entity = new TxEventoDetalles();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEventoDetalles.findById(entity.getEventoDetalleId()).orElse(new TxEventoDetalles());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEventoDetalles.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxEventoDetalles.update: " + ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxEventoDetalles dom = (DominioTxEventoDetalles) obj;  
 TxEventoDetalles entity = new TxEventoDetalles();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEventoDetalles.findById(entity.getEventoDetalleId()).orElse(new TxEventoDetalles());  
 daoTxEventoDetalles.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxEventoDetalles.delete: " + ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxEventoDetalles dom = (DominioTxEventoDetalles) obj;  
 TxEventoDetalles entity = new TxEventoDetalles();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxEventoDetalles.findById(entity.getEventoDetalleId()).orElse(new TxEventoDetalles());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 return null;  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxEventoDetalles.count();  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxEventoDetalles.count: " + ex);  
 }  
 }  
  
 @Override  
 public DominioTxEventoDetalles existsById(Object obj) throws Exception {  
 try {  
 DominioTxEventoDetalles dom = (DominioTxEventoDetalles) obj;  
 TxEventoDetalles entity = new TxEventoDetalles();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxEventoDetalles.existsById(entity.getEventoDetalleId()));  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxEventoDetalles.findById: " + ex);  
 }  
 }  
  
   
  
 @Override  
 public List<DominioTxEventoDetalles> findByTxEvento(DominioTxEvento eventoId) throws Exception {  
 try {  
 TxEvento evento = new TxEvento();  
 JRUtil.copyProperties(evento, eventoId);  
 List<DominioTxEventoDetalles> lista = new ArrayList<>();  
 List<TxEventoDetalles> listaEntities = daoTxEventoDetalles.findByEventoId(evento);  
 for (TxEventoDetalles entity : listaEntities) {  
 DominioTxEventoDetalles dom = new DominioTxEventoDetalles();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }   
 return lista;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 return null;  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxFacturaUuids.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxFacturaUuids;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxFacturaUuids;  
import mx.mbi.dgc.jpa.entity.TxFacturaUuids;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxFacturaUuids implements IfaceTxFacturaUuids {  
 @Autowired  
 private DAOTxFacturaUuids daoTxFacturaUuids;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxFacturaUuids> lista = new ArrayList<>();  
 List<TxFacturaUuids> objs = daoTxFacturaUuids.findAll();  
 for(TxFacturaUuids entity: objs) {  
 DominioTxFacturaUuids dom = new DominioTxFacturaUuids();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxFacturaUuids.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxFacturaUuids dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxFacturaUuids> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxFacturaUuids> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxFacturaUuids.findByFacturaUuidId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxFacturaUuids.findByObservacionesContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxFacturaUuids.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxFacturaUuids entity: objs) {  
 DominioTxFacturaUuids to = new DominioTxFacturaUuids();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxFacturaUuids dom = (DominioTxFacturaUuids)obj;  
 TxFacturaUuids entity = new TxFacturaUuids();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxFacturaUuids.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxFacturaUuids.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxFacturaUuids dom = (DominioTxFacturaUuids)obj;  
 TxFacturaUuids entity = new TxFacturaUuids();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxFacturaUuids.findById(entity.getFacturaUuidId()).  
 orElse(new TxFacturaUuids());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxFacturaUuids.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxFacturaUuids.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxFacturaUuids dom = (DominioTxFacturaUuids)obj;  
 TxFacturaUuids entity = new TxFacturaUuids();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxFacturaUuids.findById(entity.getFacturaUuidId()).  
 orElse(new TxFacturaUuids());  
 daoTxFacturaUuids.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxFacturaUuids.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxFacturaUuids dom = (DominioTxFacturaUuids)obj;  
 TxFacturaUuids entity = new TxFacturaUuids();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxFacturaUuids.findById(entity.getFacturaUuidId()).  
 orElse(new TxFacturaUuids());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxFacturaUuids.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxFacturaUuids.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxFacturaUuids.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxFacturaUuids existsById(Object obj) throws Exception {  
 try {  
 DominioTxFacturaUuids dom = (DominioTxFacturaUuids)obj;  
 TxFacturaUuids entity = new TxFacturaUuids();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxFacturaUuids.existsById(entity.getFacturaUuidId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxFacturaUuids.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxFormulas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxFormulas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxFormulas;  
import mx.mbi.dgc.jpa.entity.TxFormulas;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxFormulas implements IfaceTxFormulas {  
 @Autowired  
 private DAOTxFormulas daoTxFormulas;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxFormulas> lista = new ArrayList<>();  
 List<TxFormulas> objs = daoTxFormulas.findAll();  
 for(TxFormulas entity: objs) {  
 DominioTxFormulas dom = new DominioTxFormulas();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxFormulas.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxFormulas dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxFormulas> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxFormulas> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxFormulas.findByFormulaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxFormulas.findByFormulaContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxFormulas.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxFormulas entity: objs) {  
 DominioTxFormulas to = new DominioTxFormulas();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxFormulas dom = (DominioTxFormulas)obj;  
 TxFormulas entity = new TxFormulas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxFormulas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxFormulas.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxFormulas dom = (DominioTxFormulas)obj;  
 TxFormulas entity = new TxFormulas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxFormulas.findById(entity.getFormulaId()).  
 orElse(new TxFormulas());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxFormulas.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxFormulas.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxFormulas dom = (DominioTxFormulas)obj;  
 TxFormulas entity = new TxFormulas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxFormulas.findById(entity.getFormulaId()).  
 orElse(new TxFormulas());  
 daoTxFormulas.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxFormulas.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxFormulas dom = (DominioTxFormulas)obj;  
 TxFormulas entity = new TxFormulas();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxFormulas.findById(entity.getFormulaId()).  
 orElse(new TxFormulas());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxFormulas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxFormulas.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxFormulas.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxFormulas existsById(Object obj) throws Exception {  
 try {  
 DominioTxFormulas dom = (DominioTxFormulas)obj;  
 TxFormulas entity = new TxFormulas();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxFormulas.existsById(entity.getFormulaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxFormulas.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxLogErrorCarga.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxLogErrorCarga;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxLogErrorCarga;  
import mx.mbi.dgc.jpa.entity.TxLogErrorCarga;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxLogErrorCarga implements IfaceTxLogErrorCarga {  
 @Autowired  
 private DAOTxLogErrorCarga daoTxLogErrorCarga;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxLogErrorCarga> lista = new ArrayList<>();  
 List<TxLogErrorCarga> objs = daoTxLogErrorCarga.findAll();  
 for(TxLogErrorCarga entity: objs) {  
 DominioTxLogErrorCarga dom = new DominioTxLogErrorCarga();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxLogErrorCarga.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxLogErrorCarga dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxLogErrorCarga> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxLogErrorCarga> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxLogErrorCarga.findByTableName(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxLogErrorCarga.findByTableNameContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxLogErrorCarga.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxLogErrorCarga entity: objs) {  
 DominioTxLogErrorCarga to = new DominioTxLogErrorCarga();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxLogErrorCarga dom = (DominioTxLogErrorCarga)obj;  
 TxLogErrorCarga entity = new TxLogErrorCarga();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxLogErrorCarga.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxLogErrorCarga.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxLogErrorCarga dom = (DominioTxLogErrorCarga)obj;  
 TxLogErrorCarga entity = new TxLogErrorCarga();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxLogErrorCarga.findById(entity.getLogErrorCargaId()).  
 orElse(new TxLogErrorCarga());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxLogErrorCarga.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxLogErrorCarga.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxLogErrorCarga dom = (DominioTxLogErrorCarga)obj;  
 TxLogErrorCarga entity = new TxLogErrorCarga();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxLogErrorCarga.findById(entity.getLogErrorCargaId()).  
 orElse(new TxLogErrorCarga());  
 daoTxLogErrorCarga.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxLogErrorCarga.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxLogErrorCarga dom = (DominioTxLogErrorCarga)obj;  
 TxLogErrorCarga entity = new TxLogErrorCarga();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxLogErrorCarga.findById(entity.getLogErrorCargaId()).  
 orElse(new TxLogErrorCarga());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxLogErrorCarga.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxLogErrorCarga.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxLogErrorCarga.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxLogErrorCarga existsById(Object obj) throws Exception {  
 try {  
 DominioTxLogErrorCarga dom = (DominioTxLogErrorCarga)obj;  
 TxLogErrorCarga entity = new TxLogErrorCarga();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxLogErrorCarga.existsById(entity.getLogErrorCargaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxLogErrorCarga.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxPerfilesVariable.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxPerfilesVariable;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxPerfilesVariable;  
import mx.mbi.dgc.jpa.entity.TxPerfilesVariable;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxPerfilesVariable implements IfaceTxPerfilesVariable {  
 @Autowired  
 private DAOTxPerfilesVariable daoTxPerfilesVariable;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxPerfilesVariable> lista = new ArrayList<>();  
 List<TxPerfilesVariable> objs = daoTxPerfilesVariable.findAll();  
 for(TxPerfilesVariable entity: objs) {  
 DominioTxPerfilesVariable dom = new DominioTxPerfilesVariable();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxPerfilesVariable.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxPerfilesVariable dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxPerfilesVariable> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxPerfilesVariable> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxPerfilesVariable.findByPerfilId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxPerfilesVariable.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxPerfilesVariable.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxPerfilesVariable entity: objs) {  
 DominioTxPerfilesVariable to = new DominioTxPerfilesVariable();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxPerfilesVariable dom = (DominioTxPerfilesVariable)obj;  
 TxPerfilesVariable entity = new TxPerfilesVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxPerfilesVariable.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxPerfilesVariable.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxPerfilesVariable dom = (DominioTxPerfilesVariable)obj;  
 TxPerfilesVariable entity = new TxPerfilesVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxPerfilesVariable.findById(entity.getPerfilId()).  
 orElse(new TxPerfilesVariable());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxPerfilesVariable.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxPerfilesVariable.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxPerfilesVariable dom = (DominioTxPerfilesVariable)obj;  
 TxPerfilesVariable entity = new TxPerfilesVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxPerfilesVariable.findById(entity.getPerfilId()).  
 orElse(new TxPerfilesVariable());  
 daoTxPerfilesVariable.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxPerfilesVariable.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxPerfilesVariable dom = (DominioTxPerfilesVariable)obj;  
 TxPerfilesVariable entity = new TxPerfilesVariable();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxPerfilesVariable.findById(entity.getPerfilId()).  
 orElse(new TxPerfilesVariable());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxPerfilesVariable.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxPerfilesVariable.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxPerfilesVariable.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxPerfilesVariable existsById(Object obj) throws Exception {  
 try {  
 DominioTxPerfilesVariable dom = (DominioTxPerfilesVariable)obj;  
 TxPerfilesVariable entity = new TxPerfilesVariable();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxPerfilesVariable.existsById(entity.getPerfilId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxPerfilesVariable.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxProduccion.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxProduccion;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxProduccion;  
import mx.mbi.dgc.jpa.entity.TxProduccion;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxProduccion implements IfaceTxProduccion {  
 @Autowired  
 private DAOTxProduccion daoTxProduccion;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxProduccion> lista = new ArrayList<>();  
 List<TxProduccion> objs = daoTxProduccion.findAll();  
 for(TxProduccion entity: objs) {  
 DominioTxProduccion dom = new DominioTxProduccion();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxProduccion.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxProduccion dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxProduccion> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxProduccion> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxProduccion.findByProduccionId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxProduccion.findByValorContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxProduccion.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxProduccion entity: objs) {  
 DominioTxProduccion to = new DominioTxProduccion();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxProduccion dom = (DominioTxProduccion)obj;  
 TxProduccion entity = new TxProduccion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxProduccion.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxProduccion.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxProduccion dom = (DominioTxProduccion)obj;  
 TxProduccion entity = new TxProduccion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxProduccion.findById(entity.getProduccionId()).  
 orElse(new TxProduccion());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxProduccion.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxProduccion.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxProduccion dom = (DominioTxProduccion)obj;  
 TxProduccion entity = new TxProduccion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxProduccion.findById(entity.getProduccionId()).  
 orElse(new TxProduccion());  
 daoTxProduccion.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxProduccion.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxProduccion dom = (DominioTxProduccion)obj;  
 TxProduccion entity = new TxProduccion();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxProduccion.findById(entity.getProduccionId()).  
 orElse(new TxProduccion());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxProduccion.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxProduccion.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxProduccion.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxProduccion existsById(Object obj) throws Exception {  
 try {  
 DominioTxProduccion dom = (DominioTxProduccion)obj;  
 TxProduccion entity = new TxProduccion();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxProduccion.existsById(entity.getProduccionId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxProduccion.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxRolPermiso.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxRolPermiso;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxRolPermiso;  
import mx.mbi.dgc.jpa.entity.TxRolPermiso;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxRolPermiso implements IfaceTxRolPermiso {  
 @Autowired  
 private DAOTxRolPermiso daoTxRolPermiso;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxRolPermiso> lista = new ArrayList<>();  
 List<TxRolPermiso> objs = daoTxRolPermiso.findAll();  
 for(TxRolPermiso entity: objs) {  
 DominioTxRolPermiso dom = new DominioTxRolPermiso();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxRolPermiso.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxRolPermiso dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxRolPermiso> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxRolPermiso> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxRolPermiso.findByRolPermisoId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxRolPermiso.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxRolPermiso.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxRolPermiso entity: objs) {  
 DominioTxRolPermiso to = new DominioTxRolPermiso();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxRolPermiso dom = (DominioTxRolPermiso)obj;  
 TxRolPermiso entity = new TxRolPermiso();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxRolPermiso.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxRolPermiso.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxRolPermiso dom = (DominioTxRolPermiso)obj;  
 TxRolPermiso entity = new TxRolPermiso();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxRolPermiso.findById(entity.getRolPermisoId()).  
 orElse(new TxRolPermiso());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxRolPermiso.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxRolPermiso.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxRolPermiso dom = (DominioTxRolPermiso)obj;  
 TxRolPermiso entity = new TxRolPermiso();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxRolPermiso.findById(entity.getRolPermisoId()).  
 orElse(new TxRolPermiso());  
 daoTxRolPermiso.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxRolPermiso.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxRolPermiso dom = (DominioTxRolPermiso)obj;  
 TxRolPermiso entity = new TxRolPermiso();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxRolPermiso.findById(entity.getRolPermisoId()).  
 orElse(new TxRolPermiso());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxRolPermiso.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxRolPermiso.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxRolPermiso.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxRolPermiso existsById(Object obj) throws Exception {  
 try {  
 DominioTxRolPermiso dom = (DominioTxRolPermiso)obj;  
 TxRolPermiso entity = new TxRolPermiso();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxRolPermiso.existsById(entity.getRolPermisoId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxRolPermiso.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxTrazabilidad.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxTrazabilidad;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxTrazabilidad;  
import mx.mbi.dgc.jpa.entity.TxTrazabilidad;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxTrazabilidad implements IfaceTxTrazabilidad {  
 @Autowired  
 private DAOTxTrazabilidad daoTxTrazabilidad;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxTrazabilidad> lista = new ArrayList<>();  
 List<TxTrazabilidad> objs = daoTxTrazabilidad.findAll();  
 for(TxTrazabilidad entity: objs) {  
 DominioTxTrazabilidad dom = new DominioTxTrazabilidad();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxTrazabilidad.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxTrazabilidad dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxTrazabilidad> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxTrazabilidad> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxTrazabilidad.findByTrazabilidadId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxTrazabilidad.findByAccionContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxTrazabilidad.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxTrazabilidad entity: objs) {  
 DominioTxTrazabilidad to = new DominioTxTrazabilidad();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxTrazabilidad dom = (DominioTxTrazabilidad)obj;  
 TxTrazabilidad entity = new TxTrazabilidad();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxTrazabilidad.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxTrazabilidad.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxTrazabilidad dom = (DominioTxTrazabilidad)obj;  
 TxTrazabilidad entity = new TxTrazabilidad();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxTrazabilidad.findById(entity.getTrazabilidadId()).  
 orElse(new TxTrazabilidad());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxTrazabilidad.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxTrazabilidad.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxTrazabilidad dom = (DominioTxTrazabilidad)obj;  
 TxTrazabilidad entity = new TxTrazabilidad();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxTrazabilidad.findById(entity.getTrazabilidadId()).  
 orElse(new TxTrazabilidad());  
 daoTxTrazabilidad.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxTrazabilidad.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxTrazabilidad dom = (DominioTxTrazabilidad)obj;  
 TxTrazabilidad entity = new TxTrazabilidad();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxTrazabilidad.findById(entity.getTrazabilidadId()).  
 orElse(new TxTrazabilidad());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxTrazabilidad.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxTrazabilidad.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxTrazabilidad.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxTrazabilidad existsById(Object obj) throws Exception {  
 try {  
 DominioTxTrazabilidad dom = (DominioTxTrazabilidad)obj;  
 TxTrazabilidad entity = new TxTrazabilidad();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxTrazabilidad.existsById(entity.getTrazabilidadId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxTrazabilidad.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxTrazaQuery.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxTrazaQuery;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxTrazaQuery;  
import mx.mbi.dgc.jpa.entity.TxTrazaQuery;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxTrazaQuery implements IfaceTxTrazaQuery {  
 @Autowired  
 private DAOTxTrazaQuery daoTxTrazaQuery;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxTrazaQuery> lista = new ArrayList<>();  
 List<TxTrazaQuery> objs = daoTxTrazaQuery.findAll();  
 for(TxTrazaQuery entity: objs) {  
 DominioTxTrazaQuery dom = new DominioTxTrazaQuery();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxTrazaQuery.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxTrazaQuery dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxTrazaQuery> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxTrazaQuery> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxTrazaQuery.findByTrazaQueryId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxTrazaQuery.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxTrazaQuery.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxTrazaQuery entity: objs) {  
 DominioTxTrazaQuery to = new DominioTxTrazaQuery();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxTrazaQuery dom = (DominioTxTrazaQuery)obj;  
 TxTrazaQuery entity = new TxTrazaQuery();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxTrazaQuery.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxTrazaQuery.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxTrazaQuery dom = (DominioTxTrazaQuery)obj;  
 TxTrazaQuery entity = new TxTrazaQuery();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxTrazaQuery.findById(entity.getTrazaQueryId()).  
 orElse(new TxTrazaQuery());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxTrazaQuery.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxTrazaQuery.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxTrazaQuery dom = (DominioTxTrazaQuery)obj;  
 TxTrazaQuery entity = new TxTrazaQuery();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxTrazaQuery.findById(entity.getTrazaQueryId()).  
 orElse(new TxTrazaQuery());  
 daoTxTrazaQuery.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxTrazaQuery.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxTrazaQuery dom = (DominioTxTrazaQuery)obj;  
 TxTrazaQuery entity = new TxTrazaQuery();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxTrazaQuery.findById(entity.getTrazaQueryId()).  
 orElse(new TxTrazaQuery());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxTrazaQuery.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxTrazaQuery.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxTrazaQuery.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxTrazaQuery existsById(Object obj) throws Exception {  
 try {  
 DominioTxTrazaQuery dom = (DominioTxTrazaQuery)obj;  
 TxTrazaQuery entity = new TxTrazaQuery();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxTrazaQuery.existsById(entity.getTrazaQueryId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxTrazaQuery.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxUsuario.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxUsuario;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxUsuario;  
import mx.mbi.dgc.jpa.entity.TxUsuario;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxUsuario implements IfaceTxUsuario {  
 @Autowired  
 private DAOTxUsuario daoTxUsuario;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxUsuario> lista = new ArrayList<>();  
 List<TxUsuario> objs = daoTxUsuario.findAll();  
 for(TxUsuario entity: objs) {  
 DominioTxUsuario dom = new DominioTxUsuario();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuario.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxUsuario dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxUsuario> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxUsuario> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxUsuario.findByUsuarioId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxUsuario.findByUsernameContaining(""+dom.getSearch().getValue(),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxUsuario.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxUsuario entity: objs) {  
 DominioTxUsuario to = new DominioTxUsuario();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxUsuario dom = (DominioTxUsuario)obj;  
 TxUsuario entity = new TxUsuario();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxUsuario.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuario.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxUsuario dom = (DominioTxUsuario)obj;  
 TxUsuario entity = new TxUsuario();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxUsuario.findById(entity.getUsuarioId()).  
 orElse(new TxUsuario());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxUsuario.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuario.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxUsuario dom = (DominioTxUsuario)obj;  
 TxUsuario entity = new TxUsuario();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxUsuario.findById(entity.getUsuarioId()).  
 orElse(new TxUsuario());  
 daoTxUsuario.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuario.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxUsuario dom = (DominioTxUsuario)obj;  
 TxUsuario entity = new TxUsuario();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxUsuario.findById(entity.getUsuarioId()).  
 orElse(new TxUsuario());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuario.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxUsuario.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuario.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxUsuario existsById(Object obj) throws Exception {  
 try {  
 DominioTxUsuario dom = (DominioTxUsuario)obj;  
 TxUsuario entity = new TxUsuario();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxUsuario.existsById(entity.getUsuarioId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuario.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxUsuarioPlanta.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxUsuarioPlanta;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxUsuarioPlanta;  
import mx.mbi.dgc.jpa.entity.TxUsuarioPlanta;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxUsuarioPlanta implements IfaceTxUsuarioPlanta {  
 @Autowired  
 private DAOTxUsuarioPlanta daoTxUsuarioPlanta;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxUsuarioPlanta> lista = new ArrayList<>();  
 List<TxUsuarioPlanta> objs = daoTxUsuarioPlanta.findAll();  
 for(TxUsuarioPlanta entity: objs) {  
 DominioTxUsuarioPlanta dom = new DominioTxUsuarioPlanta();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuarioPlanta.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxUsuarioPlanta dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxUsuarioPlanta> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxUsuarioPlanta> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxUsuarioPlanta.findByUsuarioPlantaId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxUsuarioPlanta.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxUsuarioPlanta.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxUsuarioPlanta entity: objs) {  
 DominioTxUsuarioPlanta to = new DominioTxUsuarioPlanta();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxUsuarioPlanta dom = (DominioTxUsuarioPlanta)obj;  
 TxUsuarioPlanta entity = new TxUsuarioPlanta();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxUsuarioPlanta.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuarioPlanta.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxUsuarioPlanta dom = (DominioTxUsuarioPlanta)obj;  
 TxUsuarioPlanta entity = new TxUsuarioPlanta();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxUsuarioPlanta.findById(entity.getUsuarioPlantaId()).  
 orElse(new TxUsuarioPlanta());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxUsuarioPlanta.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuarioPlanta.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxUsuarioPlanta dom = (DominioTxUsuarioPlanta)obj;  
 TxUsuarioPlanta entity = new TxUsuarioPlanta();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxUsuarioPlanta.findById(entity.getUsuarioPlantaId()).  
 orElse(new TxUsuarioPlanta());  
 daoTxUsuarioPlanta.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuarioPlanta.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxUsuarioPlanta dom = (DominioTxUsuarioPlanta)obj;  
 TxUsuarioPlanta entity = new TxUsuarioPlanta();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxUsuarioPlanta.findById(entity.getUsuarioPlantaId()).  
 orElse(new TxUsuarioPlanta());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuarioPlanta.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxUsuarioPlanta.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuarioPlanta.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxUsuarioPlanta existsById(Object obj) throws Exception {  
 try {  
 DominioTxUsuarioPlanta dom = (DominioTxUsuarioPlanta)obj;  
 TxUsuarioPlanta entity = new TxUsuarioPlanta();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxUsuarioPlanta.existsById(entity.getUsuarioPlantaId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuarioPlanta.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\BusinessTxUsuarioRol.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxUsuarioRol;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dao.DAOTxUsuarioRol;  
import mx.mbi.dgc.jpa.entity.TxUsuarioRol;  
import mx.mbi.dgc.utils.JRUtil;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import org.springframework.stereotype.Service;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.PageRequest;  
import org.springframework.data.domain.Sort;  
  
@Service  
public class BusinessTxUsuarioRol implements IfaceTxUsuarioRol {  
 @Autowired  
 private DAOTxUsuarioRol daoTxUsuarioRol;  
  
 @Override  
 public List findAll() throws Exception {  
 try {  
 List<DominioTxUsuarioRol> lista = new ArrayList<>();  
 List<TxUsuarioRol> objs = daoTxUsuarioRol.findAll();  
 for(TxUsuarioRol entity: objs) {  
 DominioTxUsuarioRol dom = new DominioTxUsuarioRol();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuarioRol.findAll: "+ex);  
 }  
 }  
  
 @Override  
 public DominioRespuesta allPagination(DominioTxUsuarioRol dom) throws Exception {  
 DominioRespuesta respuesta = new DominioRespuesta();  
 try {  
 List<DominioTxUsuarioRol> lista = new ArrayList<>();  
 int page = Math.floorDiv(dom.getStart(), dom.getLength());  
 Page<TxUsuarioRol> objs = null;  
 if(dom.getSearch() != null && dom.getSearch().getValue() != null && (""+dom.getSearch().getValue()).trim().length() > 0) {  
 try {  
 objs = daoTxUsuarioRol.findByUsuarioRolId(new java.math.BigDecimal(""+dom.getSearch().getValue()),  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex) {  
 try {  
 objs = daoTxUsuarioRol.findAll(PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )   
 );  
 } catch(Exception ex2) {  
 }  
 }  
 } else {  
 objs = daoTxUsuarioRol.findAll(  
 PageRequest.of(  
 page, dom.getLength(), dom.getOrder().get(0).getDir().equals("asc") ?   
 Sort.Direction.ASC : Sort.Direction.DESC,   
 dom.getCampos().get(dom.getOrder().get(0).getColumn())  
 )  
 );  
 }  
 for(TxUsuarioRol entity: objs) {  
 DominioTxUsuarioRol to = new DominioTxUsuarioRol();  
 JRUtil.copyProperties(to, entity);  
 lista.add(to);  
 }  
 respuesta.setData(lista);  
 respuesta.setRecordsFiltered(objs.getTotalElements());  
 respuesta.setRecordsTotal(objs.getTotalElements());  
 respuesta.setDraw(dom.getDraw());  
 respuesta.setStart(dom.getStart());  
 respuesta.setLength(dom.getLength());  
 respuesta.setCodigo(200);  
 respuesta.setMensaje("success");  
 return respuesta;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessCatMesDiaFirma.findAllPagination: "+ex);  
 }  
 }  
  
 @Override  
 public Object save(Object obj) throws Exception {  
 try {  
 DominioTxUsuarioRol dom = (DominioTxUsuarioRol)obj;  
 TxUsuarioRol entity = new TxUsuarioRol();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxUsuarioRol.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuarioRol.save: "+ex);  
 }  
 }  
  
 @Override  
 public Object update(Object obj) throws Exception {  
 try {  
 DominioTxUsuarioRol dom = (DominioTxUsuarioRol)obj;  
 TxUsuarioRol entity = new TxUsuarioRol();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxUsuarioRol.findById(entity.getUsuarioRolId()).  
 orElse(new TxUsuarioRol());  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxUsuarioRol.save(entity);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuarioRol.update: "+ex);  
 }  
 }  
  
 @Override  
 public Object delete(Object obj) throws Exception {  
 try {  
 DominioTxUsuarioRol dom = (DominioTxUsuarioRol)obj;  
 TxUsuarioRol entity = new TxUsuarioRol();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxUsuarioRol.findById(entity.getUsuarioRolId()).  
 orElse(new TxUsuarioRol());  
 daoTxUsuarioRol.delete(entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuarioRol.delete: "+ex);  
 }  
 }  
  
 @Override  
 public Object findById(Object obj) throws Exception {  
 try {  
 DominioTxUsuarioRol dom = (DominioTxUsuarioRol)obj;  
 TxUsuarioRol entity = new TxUsuarioRol();  
 JRUtil.copyProperties(entity, dom);  
 entity = daoTxUsuarioRol.findById(entity.getUsuarioRolId()).  
 orElse(new TxUsuarioRol());  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuarioRol.findById: "+ex);  
 }  
 }  
  
 @Override  
 public long count() throws Exception {  
 try {  
 return daoTxUsuarioRol.count();  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuarioRol.count: "+ex);  
 }  
 }  
  
 @Override  
 public DominioTxUsuarioRol existsById(Object obj) throws Exception {  
 try {  
 DominioTxUsuarioRol dom = (DominioTxUsuarioRol)obj;  
 TxUsuarioRol entity = new TxUsuarioRol();  
 JRUtil.copyProperties(entity, dom);  
 dom.setFound(daoTxUsuarioRol.existsById(entity.getUsuarioRolId()));  
 return dom;  
 } catch(Exception ex) {  
 throw new Exception("Error in BusinessTxUsuarioRol.findById: "+ex);  
 }  
 }  
  
 @Override  
 public List findAllPagination(Object obj) throws Exception {  
 // TODO Implement this method  
 return Collections.emptyList();  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\carga\BusinessCargaInformacion.java

package mx.mbi.dgc.business.carga;  
  
import java.math.BigDecimal;  
  
import java.util.Map;  
  
import javax.persistence.EntityManager;  
import javax.persistence.PersistenceContext;  
  
import mx.mbi.dgc.dao.negocio.DAOCargaInformacion;  
import mx.mbi.dgc.dao.negocio.cargainicial.ICargaInformacion;  
import mx.mbi.dgc.excepcion.CargaException;  
import mx.mbi.dgc.utils.DominioTool;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCargaInformacion implements IfaceCargaInformacion {  
 private static final Logger LOG = LogManager.getLogger(BusinessCargaInformacion.class);  
 @Autowired  
 DAOCargaInformacion dao;  
   
 @Autowired  
 DominioTool dominioTool;  
   
 @PersistenceContext  
 private EntityManager em;  
   
// @Autowired  
// ICargaInformacion icargaInformacion;  
  
 @Override  
 public boolean cargaCSV(String fileName, BigDecimal idUsuario, BigDecimal idPlanta) throws CargaException {  
 try {  
 System.out.println("BusinessCargaInformacion cargaCSV idPlanta: " + idPlanta);  
 Map<String, String> mapCondiciones = dominioTool.buildCondiciones("CARGA\_INICIAL", idPlanta);  
 if(idPlanta.intValue() == 1){  
 dao.procesaFileAguila(  
 mapCondiciones.get("TABLA\_CARGA"),   
 mapCondiciones.get("FORMATO\_CARGA"),   
 fileName,   
 Integer.parseInt(mapCondiciones.get("ENCABEZADO")),   
 Integer.parseInt(mapCondiciones.get("TIPO\_CARGA")),  
 idUsuario.intValue());  
 }  
   
 if(idPlanta.intValue() == 2){  
 dao.procesaFileSol(  
 mapCondiciones.get("TABLA\_CARGA"),   
 mapCondiciones.get("FORMATO\_CARGA"),   
 fileName,   
 Integer.parseInt(mapCondiciones.get("ENCABEZADO")),   
 Integer.parseInt(mapCondiciones.get("TIPO\_CARGA")),   
 idUsuario.intValue());  
 }  
   
 return true;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke BusinessCargaInformacion.cargaCSV: " + ex);  
 throw new CargaException("Error en BusinessCargaInformacion.fileName: " + ex);  
 }  
 }  
   
 /\*  
 public boolean procesaFile(String fileName, BigDecimal idUsuario, BigDecimal idPlanta) {  
 System.out.println("BusinessCargaInformacion cargaCSV idPlanta: " + idPlanta);  
 CatPlanta catPlanta = daoCatPlanta.findById(idPlanta).orElse(new CatPlanta());  
 List<CatCondiciones> listaCondiciones = daoNegocioCatCondiciones.findAllByProcesoAndCatPlanta("CARGA\_INICIAL", catPlanta);  
   
 Map<String, String> mapCondiciones = dominioTool.buildCondiciones(listaCondiciones);  
 icargaInformacion.procesaFile(  
 mapCondiciones.get("PROCEDURE\_NAME"),   
 mapCondiciones.get("TABLA\_CARGA"),   
 mapCondiciones.get("FORMATO\_CARGA"),   
 fileName,   
 Integer.parseInt(mapCondiciones.get("ENCABEZADO")),   
 Integer.parseInt(mapCondiciones.get("TIPO\_CARGA")),   
 idUsuario.intValue());  
 return true;  
 }\*/  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\carga\BusinessCargaInformacionPerfiles.java

package mx.mbi.dgc.business.carga;  
  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.IOException;  
  
import java.math.BigDecimal;  
  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import mx.mbi.dgc.dao.DAOCatEstatusPerfil;  
import mx.mbi.dgc.dao.DAOCatPlanta;  
import mx.mbi.dgc.dao.DAOTxPerfilesVariable;  
import mx.mbi.dgc.dao.negocio.DAOVariablePerfil;  
import mx.mbi.dgc.dao.negocio.DaoNegocioCatMedidores;  
import mx.mbi.dgc.dao.negocio.DaoNegocioTxCincominutal;  
import mx.mbi.dgc.dao.negocio.DaoNegocioTxPerfilesVariable;  
import mx.mbi.dgc.dominio.DominioCargaInformacion;  
import mx.mbi.dgc.excepcion.CargaException;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatEstatusPerfil;  
import mx.mbi.dgc.jpa.entity.CatMedidores;  
import mx.mbi.dgc.jpa.entity.CatMesDiaFirma;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.jpa.entity.CatVariablePerfil;  
import mx.mbi.dgc.jpa.entity.TxCincominutal;  
import mx.mbi.dgc.jpa.entity.TxPerfilesVariable;  
import mx.mbi.dgc.utils.DominioTool;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
import org.apache.poi.ss.usermodel.Row;  
import org.apache.poi.ss.usermodel.Sheet;  
import org.apache.poi.ss.usermodel.Workbook;  
import org.apache.poi.ss.usermodel.WorkbookFactory;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCargaInformacionPerfiles implements IfaceCargaInformacionPerfiles {  
 private static final Logger LOG = LogManager.getLogger(BusinessCargaInformacionPerfiles.class);  
  
 @Autowired  
 DominioTool tool;  
  
 @Autowired  
 DaoNegocioTxCincominutal daoNegocioTxCincominutal;  
  
 @Autowired  
 DaoNegocioTxPerfilesVariable daoNegocioTxPerfilesVariable;  
  
 @Autowired  
 DAOVariablePerfil daoCatVariablePerfil;  
  
 @Autowired  
 DaoNegocioCatMedidores daoNegocioCatMedidores;  
  
 @Autowired  
 DAOCatEstatusPerfil daoCatEstatusPerfil;  
  
 @Autowired  
 DAOTxPerfilesVariable daoTxPerfilesVariable;  
  
 @Autowired  
 DAOCatPlanta daoCatPlanta;  
  
 @Autowired  
 DominioTool dominioTool;  
  
 @Override  
 public DominioCargaInformacion cargaCSV(DominioCargaInformacion retorno) throws CargaException {  
 try {  
 retorno.setStatus(true);  
 retorno.setCode(200);  
  
 CatPlanta planta = daoCatPlanta.findById(retorno.getIdPlanta()).orElse(new CatPlanta());  
 Map<String, String> mapCondiciones =  
 dominioTool.buildCondiciones("CARGA\_INICIAL\_PERFILES", planta.getPlantaId());  
  
 Map<Integer, CatVariablePerfil> mapVariablePerfil = null;  
 CatEstatusPerfil catEstatusPerfil =  
 daoCatEstatusPerfil.findById(new BigDecimal(1)).orElse(new CatEstatusPerfil()); //1 = Crudo  
 TxPerfilesVariable txPerfilesVariable = null;  
 TxCincominutal txCincominutal = null;  
 CatMesDiaFirma catMesDiaFirma = null;  
 System.out.printf("Creado cuaderno nuevo en %s\n", mapCondiciones.get("CARGA\_XLS") + retorno.getFileName());  
 FileInputStream fileInputStream = new FileInputStream(new File(mapCondiciones.get("CARGA\_XLS") + retorno.getFileName()));  
 Workbook cuaderno = WorkbookFactory.create(fileInputStream);  
  
 mapVariablePerfil = catalogoVariables(cuaderno, planta);  
 Sheet hssfSheet = null;  
 Row hssfRow = null;  
 int cols = 0;  
 String nombreMedidor;  
 List<CatMedidores> listaCatMedidores = null;  
 listaCatMedidores = daoNegocioCatMedidores.findAllByCatPlanta(planta);  
  
 for (int i = 1; i < cuaderno.getNumberOfSheets(); i++) {  
 hssfSheet = cuaderno.getSheetAt(i);  
 nombreMedidor = hssfSheet.getSheetName();  
 CatMedidores catMedidor = null;  
 for (CatMedidores catMedidores : listaCatMedidores) {  
 if (nombreMedidor.indexOf(catMedidores.getNombre()) != -1) {  
 catMedidor = catMedidores;  
 break;  
 }  
 }  
 int rows = hssfSheet.getLastRowNum();  
 for (int r = 1; r < rows; r++) {  
 hssfRow = hssfSheet.getRow(r);  
 if (hssfRow == null) {  
 continue;  
 }  
 try {  
 txCincominutal = null;  
 if (hssfRow.getCell(0) == null) {  
 //TODO: Aqui hay que poner logica de negocio segun el cincominutla faltante  
 continue;  
 }  
 //System.out.println("hssfRow.getCell(0).getDateCellValue().toString(): " + hssfRow.getCell(0).getCellTypeEnum());  
 catMesDiaFirma =  
 tool.buildingCatMesDiaFirma("dd/MM/yyyy hh:mm:ss.SSS a",  
 hssfRow.getCell(0).getStringCellValue(), planta.getPlantaId());  
 if (catMesDiaFirma != null) {  
 txCincominutal = daoNegocioTxCincominutal.findByCatMesDiaFirma(catMesDiaFirma);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 if (txCincominutal == null) {  
 System.out.println("olas del mar no existe cincominutal para: ");  
 System.out.println("Buscar Cincominutal por firma: " +  
 catMesDiaFirma.getCatFirmaCincominutal().getFirma() + " y planta: " +  
 catMesDiaFirma.getCatMesDia().getDia() + "/" + catMesDiaFirma.getCatMesDia()  
 .getCatAnioMes()  
 .getAnioMes() +  
 " " + catMesDiaFirma.getCatMesDia()  
 .getCatAnioMes()  
 .getCatPlanta()  
 .getPlantaId() + " : " + catMesDiaFirma.getCatMesDia()  
 .getCatAnioMes()  
 .getCatPlanta()  
 .getNombre());  
 continue;  
 }  
 List<TxPerfilesVariable> listaVariablesPerfil = null;  
 try {  
 listaVariablesPerfil =  
 daoNegocioTxPerfilesVariable.findByTxCincominutalAndCatMedidoresAndPrincipal(txCincominutal,  
 catMedidor,  
 new BigDecimal(1));  
 } catch (DyCNormaException e) {  
 e.printStackTrace();  
 }  
 if (listaVariablesPerfil == null || listaVariablesPerfil.isEmpty()) {  
 cols = hssfRow.getLastCellNum();  
 for (int c = 1; c < cols; c++) {  
 txPerfilesVariable = new TxPerfilesVariable();  
 txPerfilesVariable.setCatVariablePerfil((CatVariablePerfil) mapVariablePerfil.get(c));  
 txPerfilesVariable.setCatMedidores(catMedidor);  
 txPerfilesVariable.setValor(new BigDecimal(hssfRow.getCell(c).getNumericCellValue()));  
 txPerfilesVariable.setPrincipal(new BigDecimal(1));  
 txPerfilesVariable.setTxCincominutal(txCincominutal);  
 txPerfilesVariable.setCatEstatusPerfil(catEstatusPerfil);  
 daoTxPerfilesVariable.save(txPerfilesVariable);  
 }  
 }  
 }  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return retorno;  
 }  
  
 private Map<Integer, CatVariablePerfil> catalogoVariables(Workbook hssfWorkbook, CatPlanta planta) {  
 Map<Integer, CatVariablePerfil> retorno = new HashMap<Integer, CatVariablePerfil>();  
 Row hssfRow = null;  
 Sheet primeraHoja = hssfWorkbook.getSheetAt(1);  
 hssfRow = primeraHoja.getRow(0);  
 int cols = hssfRow.getLastCellNum();  
 List<CatVariablePerfil> listCatVariablePerfil = null;  
 try {  
 listCatVariablePerfil = daoCatVariablePerfil.findAllByCatPlanta(planta);  
 } catch (DyCNormaException e) {  
 e.printStackTrace();  
 }  
 for (int c = 1; c < cols; c++) {  
 for (CatVariablePerfil catVariablePerfil : listCatVariablePerfil) {  
 if (hssfRow.getCell(c).getStringCellValue().equals(catVariablePerfil.getVariablePerfil())) {  
 retorno.put(c, catVariablePerfil);  
 System.out.println(c + ":" + catVariablePerfil.getVariablePerfil() + ":" + catVariablePerfil.getCatPlanta().getPlantaId());  
 break;  
 }  
 }  
 }  
 return retorno;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\carga\IfaceCargaInformacion.java

package mx.mbi.dgc.business.carga;  
  
import java.math.BigDecimal;  
  
import mx.mbi.dgc.excepcion.CargaException;  
  
public interface IfaceCargaInformacion {  
 boolean cargaCSV(String fileName, BigDecimal idUsuario, BigDecimal idPlanta) throws CargaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\carga\IfaceCargaInformacionPerfiles.java

package mx.mbi.dgc.business.carga;  
  
import mx.mbi.dgc.dominio.DominioCargaInformacion;  
import mx.mbi.dgc.excepcion.CargaException;  
  
public interface IfaceCargaInformacionPerfiles {  
 DominioCargaInformacion cargaCSV(DominioCargaInformacion dominio) throws CargaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\dyc\BusinessCorreccionNorma.java

package mx.mbi.dgc.business.dyc;  
  
import java.math.BigDecimal;  
  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.List;  
import java.util.concurrent.Future;  
  
import mx.mbi.dgc.dao.DAOCatPlanta;  
import mx.mbi.dgc.dao.negocio.DAOVariable;  
import mx.mbi.dgc.dominio.DominioDyCNorma;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatEstatusVariable;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.jpa.entity.CatVariable;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.beans.factory.annotation.Qualifier;  
import org.springframework.context.ApplicationContext;  
import org.springframework.core.task.AsyncTaskExecutor;  
import org.springframework.stereotype.Service;  
  
  
@Service  
public class BusinessCorreccionNorma implements IfaceCorreccionNorma {  
 private static final Logger LOG = LogManager.getLogger(BusinessCorreccionNorma.class);  
  
 @Autowired  
 private DAOVariable daoCatVariable;  
 @Autowired  
 private DAOCatPlanta daoCatPlanta;  
  
 @Autowired  
 @Qualifier("simpleAsyncTaskExecutor")  
 private AsyncTaskExecutor asyncTaskExecutor;  
  
 @Autowired  
 private ApplicationContext applicationContext;  
  
 @Autowired  
 private TaskCorreccionAmbientales taskCorreccionAmbientales;  
  
 CatPlanta catPlanta;  
 CatEstatusVariable catEstatusVariable;  
 CatEstatusVariable catEstatusVarNoError;  
 BigDecimal valido = new BigDecimal("27"); //28 = VALIDO  
 BigDecimal validoSinCorregir = new BigDecimal("28"); //28 = VALIDO SIN CORREGIR  
 BigDecimal noCorregido = new BigDecimal("-1"); //-1 = NO CORREGIDO  
  
 @Override  
 public DominioDyCNorma corrigeErrores(DominioDyCNorma dom) throws DyCNormaException {  
 System.out.println("#### corrigeErrores execute: " + new Date());  
 DominioDyCNorma response = new DominioDyCNorma();  
 response.setStatus(false);  
 response.setCode(500);  
 try {  
 catPlanta = daoCatPlanta.findById(dom.getCatPlanta().getPlantaId()).orElse(new CatPlanta());  
 response = corrigeAmbientales(dom);  
 if (response.getCode() == 200) {  
 response = corrigeCromatrograficas(dom);  
 } else {  
 response.setStatus(false);  
 response.setCode(500);  
 response.setMsg("Hubo un error al corregir variables Ambientales.");  
 }  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en BusinessCorreccionNorma.corrigeErrores: " + ex);  
 }  
 return response;  
 }  
  
 private DominioDyCNorma corrigeAmbientales(DominioDyCNorma dom) throws DyCNormaException {  
  
 //TODO: Intentar secuencial en lugar de usando hilos  
 try {  
 List<CatVariable> variables = daoCatVariable.findAllByTipoAndCatPlanta("A", catPlanta);  
  
 //List<Future<?>> futureList = new ArrayList<>();  
  
 for (CatVariable tipoDeVariable : variables) {  
 System.out.printf("Realizando Correccion de Norma de %s\n", tipoDeVariable.getNombre());  
 taskCorreccionAmbientales.setCatVariable(tipoDeVariable);  
 taskCorreccionAmbientales.setCatPlanta(catPlanta);  
 taskCorreccionAmbientales.run();  
 }  
 /\*  
 for (CatVariable varCat : variables) {  
 taskCorreccionAmbientales = applicationContext.getBean(TaskCorreccionAmbientales.class);  
 taskCorreccionAmbientales.setCatVariable(varCat);  
 Future<?> future = asyncTaskExecutor.submit(taskCorreccionAmbientales);  
 //futureList contiene un future por cada CatVariable  
 futureList.add(future);  
 }  
  
 for (Future<?> future : futureList) {  
 while(!future.isDone()) {  
 try {  
 Thread.sleep(300);  
 } catch (InterruptedException e) {  
 }  
 }  
 }  
 \*/  
 System.out.println("Todos los hilos TaskCorreccionAmbientales terminaron su proceso.");  
   
   
   
 dom.setStatus(true);  
 dom.setCode(200);  
   
 dom = corrigeCromatrograficas(dom);  
   
 return dom;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en BusinessCorreccionNorma.corrigeAmbientales: " + ex);  
 }  
 }  
  
 private DominioDyCNorma corrigeCromatrograficas(DominioDyCNorma dom) throws DyCNormaException {  
   
 try {  
 List<CatVariable> variables = daoCatVariable.findAllByTipoAndCatPlanta("C", catPlanta);  
 TaskCorreccionCromatrograficas taskCorreccionCromatrograficas =  
 applicationContext.getBean(TaskCorreccionCromatrograficas.class);  
 //List<Future<?>> futureList = new ArrayList<>();  
  
 for (CatVariable tipoDeVariable : variables) {  
 System.out.printf("Realizando Correccion de Cromatograficas de %s\n", tipoDeVariable.getNombre());  
 taskCorreccionCromatrograficas.setCatVariable(tipoDeVariable);  
 taskCorreccionCromatrograficas.setCatPlanta(catPlanta);  
 taskCorreccionCromatrograficas.run();  
 }  
  
 //Future<?> future = asyncTaskExecutor.submit(taskCorreccionCromatrograficas);  
   
 //futureList.add(future);  
  
// for (Future<?> futuredos : futureList) {  
// while(!futuredos.isDone()) {  
// try {  
// Thread.sleep(300);  
// } catch (InterruptedException e) {  
// }  
// }  
// }  
 System.out.println("Todos los hilos TaskCorreccionCromatrograficas terminaron su proceso.");  
   
   
   
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en BusinessCorreccionNorma.corrigeCromatrograficas: " + ex);  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\dyc\BusinessCorreccionNormaPerfiles.java

package mx.mbi.dgc.business.dyc;  
  
import java.math.BigDecimal;  
  
import java.text.DateFormat;  
import java.text.SimpleDateFormat;  
  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import mx.mbi.dgc.dao.DAOCatEstatusPerfil;  
import mx.mbi.dgc.dao.DAOCatPlanta;  
import mx.mbi.dgc.dao.DAOTxPerfilesVariable;  
import mx.mbi.dgc.dao.negocio.DAOAnioMes;  
import mx.mbi.dgc.dao.negocio.DAODyCNorma;  
import mx.mbi.dgc.dao.negocio.DAOMesDia;  
import mx.mbi.dgc.dao.negocio.DAOMesDiaFirma;  
import mx.mbi.dgc.dao.negocio.DAOVariablePerfil;  
import mx.mbi.dgc.dao.negocio.DaoNegocioCatMedidores;  
import mx.mbi.dgc.dao.negocio.DaoNegocioTxCincominutal;  
import mx.mbi.dgc.dao.negocio.DaoNegocioTxPerfilesVariable;  
import mx.mbi.dgc.dominio.DominioCatEstatusVariable;  
import mx.mbi.dgc.dominio.DominioCatVariable;  
import mx.mbi.dgc.dominio.DominioDyCNorma;  
import mx.mbi.dgc.dominio.DominioReporteDyC;  
import mx.mbi.dgc.dominio.DominioTags;  
import mx.mbi.dgc.dominio.Evaluacion;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatAnioMes;  
import mx.mbi.dgc.jpa.entity.CatEstatusPerfil;  
import mx.mbi.dgc.jpa.entity.CatMedidores;  
import mx.mbi.dgc.jpa.entity.CatMesDia;  
import mx.mbi.dgc.jpa.entity.CatMesDiaFirma;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.jpa.entity.CatVariablePerfil;  
import mx.mbi.dgc.jpa.entity.TxCincominutal;  
import mx.mbi.dgc.jpa.entity.TxPerfilesVariable;  
import mx.mbi.dgc.utils.DominioTool;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
  
@Service  
public class BusinessCorreccionNormaPerfiles implements IfaceCorreccionNormaPerfiles {  
 private static final Logger LOG = LogManager.getLogger(BusinessCorreccionNormaPerfiles.class);  
  
 @Autowired  
 private DAOVariablePerfil daoCatVariablePerfil;  
  
 @Autowired  
 private DaoNegocioTxPerfilesVariable daoNegocioTxPerfilesVariable;  
  
 @Autowired  
 private DAOCatEstatusPerfil daoCatEstatusPerfil;  
  
 @Autowired  
 private DAOTxPerfilesVariable daoTxPerfilesVariable;  
  
 @Autowired  
 DominioTool tool;  
  
 @Autowired  
 DaoNegocioTxCincominutal daoNegocioTxCincominutal;  
  
 @Autowired  
 private DaoNegocioCatMedidores daoNegocioCatMedidores;  
  
 @Autowired  
 private IfaceDyCNorma businessIfaceDyCNorma;  
  
  
 @Autowired  
 private DAODyCNorma daoDyCNorma;  
 @Autowired  
 private DAOMesDiaFirma daoMesDiaFirma;  
 @Autowired  
 private DAOMesDia daoMesDia;  
 @Autowired  
 private DAOAnioMes daoAnioMes;  
 @Autowired  
 private DAOCatPlanta daoCatPlanta;  
  
 CatPlanta catPlanta;  
 TxPerfilesVariable varUltimaValida;  
 Evaluacion evalExitosa;  
 CatEstatusPerfil catEstatusVariable;  
  
 CatEstatusPerfil catEstatusVarNoError;  
 int valido = 2;  
 int sinCorregir = 3;  
 BigDecimal activo = new BigDecimal("1");  
 BigDecimal inactivo = new BigDecimal("0");  
  
 @Override  
 public DominioDyCNorma corrigeErrores(DominioDyCNorma dom) throws DyCNormaException {  
 DominioDyCNorma response = new DominioDyCNorma();  
 response.setStatus(false);  
 response.setCode(500);  
 try {  
 catPlanta = daoCatPlanta.findById(dom.getCatPlanta().getPlantaId()).orElse(new CatPlanta());  
 catEstatusVariable =  
 daoCatEstatusPerfil.findById(new BigDecimal(2)).orElse(new CatEstatusPerfil()); //2 = VALIDO  
 catEstatusVarNoError =  
 daoCatEstatusPerfil.findById(new BigDecimal(3))  
 .orElse(new CatEstatusPerfil()); //3 = VALIDO SIN CORREGIR  
 response = corrigePerfiles(dom);  
 if (response.getCode() == 200) {  
 System.out.println("correcion#################################################################");  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } else {  
 response.setStatus(false);  
 response.setCode(500);  
 response.setMsg("Hubo un error al corregir variables PERFILES.");  
 }  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en BusinessCorreccionNorma.corrigeErrores: " + ex);  
 }  
 return response;  
 }  
  
 private DominioDyCNorma corrigePerfiles(DominioDyCNorma dom) throws DyCNormaException {  
 List<TxPerfilesVariable> tuplaCorreccion = new ArrayList<TxPerfilesVariable>();  
 try {  
 for (CatMedidores catMedidor : daoNegocioCatMedidores.findAllByCatPlanta(catPlanta)) {  
 for (CatVariablePerfil catVariable : daoCatVariablePerfil.findAllByCatPlanta(catPlanta)) {  
 tuplaCorreccion = new ArrayList<TxPerfilesVariable>();  
 varUltimaValida =  
 daoNegocioTxPerfilesVariable.getAnclaValidaCorreccion(catVariable.getVariablePerfilId(),  
 catPlanta.getPlantaId(),  
 catMedidor.getMedidorId());  
 if (varUltimaValida == null) {  
 continue;  
 }  
 List<TxPerfilesVariable> listaTxPerfilesVariable = null;  
 try {  
 listaTxPerfilesVariable =  
 daoNegocioTxPerfilesVariable.getListTxPerfilesVariable(  
 varUltimaValida.getTxCincominutal().getCincominutalId(),  
 varUltimaValida.getTxCincominutal()  
 .getCatMesDiaFirma()  
 .getCatMesDia()  
 .getCatAnioMes()  
 .getAnioMesId(),  
 varUltimaValida.getTxCincominutal()  
 .getCatMesDiaFirma()  
 .getCatMesDia()  
 .getMesDiaId(),  
 varUltimaValida.getTxCincominutal()  
 .getCatMesDiaFirma()  
 .getCatFirmaCincominutal()  
 .getFirmaId(),  
 catVariable  
 .getVariablePerfilId(),  
 dom  
 .getCatPlanta()  
 .getPlantaId(),  
 catMedidor  
 .getMedidorId());  
 } catch (Exception ex) {  
 listaTxPerfilesVariable =  
 daoNegocioTxPerfilesVariable.getAllTxPerfilesVariable(catVariable.getVariablePerfilId(),  
 dom.getCatPlanta().getPlantaId(),  
 catMedidor.getMedidorId());  
 }  
  
 tuplaCorreccion.add(varUltimaValida);  
 for (TxPerfilesVariable var : listaTxPerfilesVariable) {  
 tuplaCorreccion.add(var);  
 if (!var.getCatEstatusPerfil()  
 .getEstatusPerfilId()  
 .equals(catEstatusVariable.getEstatusPerfilId())) {  
 continue;  
 }  
 if (var.getCatEstatusPerfil()  
 .getEstatusPerfilId()  
 .equals(catEstatusVariable.getEstatusPerfilId())) { //es valida  
 if (tuplaCorreccion.size() == 2) {  
 persistirTxPerfilesVariable(tuplaCorreccion.get(0), catEstatusVarNoError,  
 tuplaCorreccion.get(0).getValor());  
 tuplaCorreccion = new ArrayList<TxPerfilesVariable>();  
 tuplaCorreccion.add(var);  
 continue;  
 }  
 List<TxPerfilesVariable> corregidos = corregirAmbientales(tuplaCorreccion, dom);  
  
 tuplaCorreccion = new ArrayList<TxPerfilesVariable>();  
 tuplaCorreccion.add(var);  
 }  
 }  
 }  
 }  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en BusinessCorreccionNorma.corrigeAmbientales: " + ex);  
 }  
 }  
  
 private void persistirTxPerfilesVariable(TxPerfilesVariable txCVar, CatEstatusPerfil catEstatusPerfil,  
 BigDecimal valor) {  
 TxPerfilesVariable nueva = new TxPerfilesVariable();  
 nueva.setCatMedidores(txCVar.getCatMedidores());  
 nueva.setCatVariablePerfil(txCVar.getCatVariablePerfil());  
 nueva.setPrincipal(activo);  
 txCVar.setPrincipal(inactivo);  
 nueva.setTxCincominutal(txCVar.getTxCincominutal());  
 nueva.setValor(valor);  
 nueva.setCatEstatusPerfil(catEstatusPerfil);  
 daoTxPerfilesVariable.save(txCVar);  
 daoTxPerfilesVariable.save(nueva);  
 }  
  
 private List<TxPerfilesVariable> corregirAmbientales(List<TxPerfilesVariable> lista,  
 DominioDyCNorma dom) throws Exception {  
 TxPerfilesVariable varPrimeraValida = lista.get(0);  
 TxPerfilesVariable varB = lista.get(lista.size() - 1);  
  
 persistirTxPerfilesVariable(varPrimeraValida, catEstatusVarNoError, varPrimeraValida.getValor());  
 lista.remove(0);  
 lista.remove(lista.size() - 1);  
  
 int errores = lista.size();  
 if (errores > 0 && errores < 13) {  
 sustitucionMedidorRespaldo(lista);  
 }  
 if (errores > 12 && errores < 61) {  
 sustitucionAnterior(lista, 2, 1);  
 }  
 if (errores > 60) {  
 interpolacionAmbientales(lista, varPrimeraValida, varB);  
 }  
  
 return lista;  
 }  
  
 private void interpolacionAmbientales(List<TxPerfilesVariable> lista, TxPerfilesVariable varPrimeraValida,  
 TxPerfilesVariable varB) throws Exception {  
 BigDecimal a = varPrimeraValida.getValor();  
 BigDecimal b = varB.getValor();  
 int x = 1;  
 int n = lista.size();  
 for (TxPerfilesVariable var : lista) {  
 double valor = JRUtil.interpolacion(x, n, a, b);  
 x++;  
 persistirTxPerfilesVariable(var,  
 daoCatEstatusPerfil.findById(new BigDecimal(24)).orElse(new CatEstatusPerfil()), //24 = CORREGIDO POR INTERPOLACION LINEAL  
 new BigDecimal(valor));  
 }  
  
 }  
  
 private void sustitucionMedidorRespaldo(List<TxPerfilesVariable> lista) throws Exception {  
 CatMedidores medidorSecundario = null;  
 CatMesDiaFirma pasado = null;  
 for (TxPerfilesVariable txVarStatus : lista) {  
 if (txVarStatus.getCatMedidores()  
 .getNombre()  
 .indexOf("M2") != -1) {  
 continue;  
 }  
 medidorSecundario = getMedidorSecundario(txVarStatus.getCatMedidores());  
 TxPerfilesVariable TxPerfilesVariableMedidorSecundario =  
 daoNegocioTxPerfilesVariable.findByTxCincominutalAndCatMedidoresAndCatVariablePerfilAndPrincipal(txVarStatus.getTxCincominutal(),  
 medidorSecundario,  
 txVarStatus.getCatVariablePerfil(),  
 new BigDecimal(1));  
 if (TxPerfilesVariableMedidorSecundario.getCatEstatusPerfil()  
 .getEstatusPerfilId()  
 .equals(catEstatusVariable.getEstatusPerfilId())) {  
 persistirTxPerfilesVariable(txVarStatus,  
 daoCatEstatusPerfil.findById(new BigDecimal(28))  
 .orElse(new CatEstatusPerfil()), //28 = CORREGIDO POR MEDIDOR DE RESPALDO TODO  
 TxPerfilesVariableMedidorSecundario.getValor());  
 }  
  
 if (!TxPerfilesVariableMedidorSecundario.getCatEstatusPerfil()  
 .getEstatusPerfilId()  
 .equals(catEstatusVariable.getEstatusPerfilId())) {  
 persistirTxPerfilesVariable(txVarStatus,  
 daoCatEstatusPerfil.findById(new BigDecimal(29))  
 .orElse(new CatEstatusPerfil()), //28 = CORREGIDO POR MEDIDOR DEL PRODUCTOR TODO  
 TxPerfilesVariableMedidorSecundario.getValor());  
 }  
 }  
 }  
  
 private Map<String, CatMedidores> medidoresSecundarios = new HashMap<String, CatMedidores>();  
  
 private CatMedidores getMedidorSecundario(CatMedidores medidorPrimario) {  
 String nombreMedidorSecundario = medidorPrimario.getNombre().replaceAll("M1", "M2");  
 CatMedidores retorno = medidoresSecundarios.get(nombreMedidorSecundario);  
  
 if (retorno == null) {  
 try {  
 retorno =  
 daoNegocioCatMedidores.findByNombreAndCatPlanta(nombreMedidorSecundario,  
 medidorPrimario.getCatPlanta());  
 medidoresSecundarios.put(nombreMedidorSecundario, retorno);  
 } catch (DyCNormaException e) {  
 }  
 }  
 return retorno;  
 }  
  
 private void sustitucionAnterior(List<TxPerfilesVariable> lista, Integer unidadTiempo,  
 Integer saltoTemporal) throws Exception {  
 CatMesDiaFirma pasado = null;  
 CatEstatusPerfil catEstatusPerfil = null;  
  
 switch (unidadTiempo) {  
 case 1: //cincominutal  
 catEstatusPerfil =  
 daoCatEstatusPerfil.findById(new BigDecimal(25))  
 .orElse(new CatEstatusPerfil()); //27 = CORREGIDO POR SUSTITUCION CINCOMINUTAL ANTERIOR  
 break;  
 case 2: //mes  
 catEstatusPerfil =  
 daoCatEstatusPerfil.findById(new BigDecimal(25))  
 .orElse(new CatEstatusPerfil()); //26 = CORREGIDO POR SUSTITUCION MES ANTERIOR  
 break;  
 case 3: //dia  
 catEstatusPerfil =  
 daoCatEstatusPerfil.findById(new BigDecimal(25))  
 .orElse(new CatEstatusPerfil()); //25 = CORREGIDO POR SUSTITUCION DIA ANTERIOR  
 break;  
 default:  
 }  
  
  
 for (TxPerfilesVariable txPerfilesVariable : lista) {  
 pasado =  
 tool.getMinusCatMesDiaFirma(txPerfilesVariable.getTxCincominutal().getCatMesDiaFirma(), unidadTiempo,  
 saltoTemporal);  
 TxPerfilesVariable TxPerfilesVariablePasada =  
 daoNegocioTxPerfilesVariable.getTxPerfilesVariableByCatMesDiaFirma(txPerfilesVariable.getCatVariablePerfil().getVariablePerfilId(),  
 txPerfilesVariable.getTxCincominutal().getCatPlanta().getPlantaId(),  
 txPerfilesVariable.getCatMedidores().getMedidorId(),  
 pasado.getMesDiaFirmaId());  
 persistirTxPerfilesVariable(txPerfilesVariable, catEstatusPerfil, TxPerfilesVariablePasada.getValor());  
 }  
 }  
  
  
 private List<TxPerfilesVariable> sustitucionPorPlantaAmbientales(List<TxPerfilesVariable> lista) throws Exception {  
 List<TxPerfilesVariable> sustituciones = new ArrayList<>();  
 CatPlanta nuevaPlanta = null;  
 if (catPlanta.getPlantaId().intValue() == 1) {  
 nuevaPlanta = daoCatPlanta.findById(new BigDecimal(2)).orElse(new CatPlanta());  
 }  
 if (catPlanta.getPlantaId().intValue() == 2) {  
 nuevaPlanta = daoCatPlanta.findById(new BigDecimal(1)).orElse(new CatPlanta());  
 }  
 CatAnioMes catAnioMes = daoAnioMes.findByAnioMesAndCatPlanta("anioMes", nuevaPlanta);  
 int ultimoDia = JRUtil.getUltimoDiaMesAnterior();  
 for (int i = ultimoDia; i > 0; i--) {  
 CatMesDia mesDia = daoMesDia.findByCatAnioMesAndDia(catAnioMes, new BigDecimal(i));  
 List<CatMesDiaFirma> firmas = daoMesDiaFirma.findAllByCatMesDia(mesDia);  
 for (CatMesDiaFirma catMesDiaFirma : firmas) {  
 LOG.info("######### sustitucionMesAnterior catMesDiaFirma firma Id: " +  
 catMesDiaFirma.getMesDiaFirmaId());  
 TxCincominutal ccm = daoDyCNorma.findByCatMesDiaFirma(catMesDiaFirma);  
 if (ccm == null) {  
 System.out.println("correccion No existen cincominutales para la planta: " +  
 nuevaPlanta.getNombre());  
 return sustituciones;  
 }  
 LOG.info("correccion######### sustitucionMesAnterior TxCincominutal Id: " + ccm.getCincominutalId());  
 /\*for (TxCincominutalVar var : ccm.getTxCincominutalVars()) {  
 for (TxPerfilesVariable txVarStatus : lista) {  
 if (txVarStatus.getTxCincominutal()  
 .getCatMesDiaFirma()  
 .getCatFirmaCincominutal()  
 .getFirma()  
 .equals(var.getTxCincominutal()  
 .getCatMesDiaFirma()  
 .getCatFirmaCincominutal()  
 .getFirma())) {  
 TxPerfilesVariable varSustituir = txVarStatus;  
 varSustituir.setValor(var.getValorNorma());  
 //varSustituir.setCatEstatusVariable(daoCatEstatusVariable.findById(new BigDecimal(24))  
 // .orElse(new CatEstatusVariable())); //24 = CORREGIDO POR SUSTITUCION VALORES OTRA PLANTA  
 sustituciones.add(varSustituir);  
 }  
 }  
 }\*/  
 }  
 }  
 return sustituciones;  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/getTags", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioReporteDyC getTags(@RequestBody DominioDyCNorma dom) throws Exception {  
 DominioReporteDyC tags = new DominioReporteDyC();  
 tags.setStatus(false);  
 tags.setCode(500);  
 try {  
 LOG.info("invoke RESTDyCNorma.getTags. . .");  
 tags = businessIfaceDyCNorma.getTags(dom);  
 LOG.info("####### tags size(): " + tags.getTags().size());  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTDyCNorma.getTags: " + ex);  
 dom.setMsg("Error in invoke RESTDyCNorma.getTags: " + ex);  
 }  
 return tags;  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/getValoresTags", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioReporteDyC getValoresTags(@RequestBody DominioReporteDyC tags) throws Exception {  
 System.out.println("correccion################################ Tags Values: " + tags.getValuesTags());  
 DominioReporteDyC reporte = new DominioReporteDyC();  
 reporte.setStatus(false);  
 reporte.setCode(500);  
 Map<String, List<String>> map = new HashMap<>();  
 for (int key = 1; key < 13; key++) {  
 if (key != 9 && key != 11) {  
 map.put("" + key, new ArrayList<>());  
 }  
 }  
 try {  
 String tagsValues = tags.getValuesTags();  
 tagsValues = tagsValues.substring(0, tagsValues.length() - 1);  
 //System.out.println("######## Values Tags: "+tagsValues);  
 //System.out.println("######## Planta Id from REST: "+tags.getPlanta().getPlantaId());  
 String[] valuesTags = tags.getValuesTags().split("\\|");  
 for (String val : valuesTags) {  
 if (val.contains("\_")) {  
 String[] idVarStatus = val.split("\\\_");  
 String key = idVarStatus[0];  
 String status = idVarStatus[1];  
 map.get(key).add(status);  
 //System.out.println("####### Mapa add key: "+key+" - Dato: "+status+" Size mapa: "+map.get(key).size());  
 }  
 }  
 for (String key : map.keySet()) {  
 DominioTags tag = new DominioTags();  
 DominioCatVariable variable = new DominioCatVariable();  
 variable.setVariableId(new BigDecimal(key));  
 tag.setVariable(variable);  
 List<String> estatuses = map.get(key);  
 //System.out.println("####### estatuses mapa size() key - Var id: "+key+" - "+estatuses.size());  
 if (estatuses != null && estatuses.size() > 0) {  
 for (String status : estatuses) {  
 DominioCatEstatusVariable varStatus = new DominioCatEstatusVariable();  
 varStatus.setEstatusVarId(new BigDecimal(status));  
 tag.getEstatuses().add(varStatus);  
 }  
 // tags.getTags().add(tag);  
 }  
 }  
 if (tags.getRangosFechas() != null && tags.getRangosFechas().length() > 0) {  
 String[] rangosFechas = tags.getRangosFechas()  
 .trim()  
 .split("\\-");  
 DateFormat df = new SimpleDateFormat("MM/dd/yyyy");  
 Date date = new Date();  
 date = df.parse(rangosFechas[0]);  
 tags.setFechaInicio(date);  
 date = df.parse(rangosFechas[1]);  
 tags.setFechaFin(date);  
 }  
 System.out.println("correccion##### Fecha de Inicio: " + tags.getFechaInicio());  
 System.out.println("correccion##### Fecha de Fin: " + tags.getFechaFin());  
 LOG.info("invoke RESTDyCNorma.getTags. . .");  
 reporte = businessIfaceDyCNorma.getValoresTags(tags);  
 System.out.println("correccion####### tags size(): " + reporte.getTags().size());  
 LOG.info("correcccion####### tags size(): " + reporte.getTags().size());  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTDyCNorma.getTags: " + ex);  
 reporte.setMsg("Error in invoke RESTDyCNorma.getTags: " + ex);  
 }  
 return reporte;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\dyc\BusinessDeteccionNorma.java

package mx.mbi.dgc.business.dyc;  
  
import java.io.File;  
  
  
import java.lang.reflect.Array;  
import java.math.BigDecimal;  
  
  
import java.sql.\*;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
  
import java.util.\*;  
import java.util.Date;  
import java.util.concurrent.Future;  
  
  
import com.mysql.jdbc.jdbc2.optional.MysqlDataSource;  
import mx.mbi.dgc.dao.DAOCatEstatusPerfil;  
import mx.mbi.dgc.dao.DAOCatEstatusVariable;  
import mx.mbi.dgc.dao.DAOCatPlanta;  
import mx.mbi.dgc.dao.DAOCatTags;  
import mx.mbi.dgc.dao.negocio.DAOVariable;  
import mx.mbi.dgc.dao.negocio.DaoNegocioTxCincominutalVar;  
import mx.mbi.dgc.dao.negocio.DaoNegocioTxPerfilesVariable;  
import mx.mbi.dgc.dominio.DominioDyCNorma;  
import mx.mbi.dgc.jpa.entity.\*;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.global.Globales;  
  
import mx.mbi.dgc.jpa.entity.TxCincominutalVar;  
  
import mx.mbi.dgc.utils.exportadorexcel.ExportadorExcel;  
  
import oracle.jdbc.OracleResultSetMetaData;  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.beans.factory.annotation.Qualifier;  
import org.springframework.beans.factory.annotation.Value;  
import org.springframework.context.ApplicationContext;  
import org.springframework.core.task.AsyncTaskExecutor;  
import org.springframework.stereotype.Service;  
import org.springframework.web.context.annotation.ApplicationScope;  
  
@ApplicationScope  
@Service  
public class BusinessDeteccionNorma implements IfaceDeteccionNorma {  
 private static final Logger LOG = LogManager.getLogger(BusinessDeteccionNorma.class);  
 private static final String NOMBRE\_ARCHIVO\_F = "Formato.xlsx";  
 private static final String NOMBRE\_ARCHIVO\_N = "Norma.xlsx";  
 private static final String NOMBRE\_ARCHIVO\_PER = "Perfiles.xlsx";  
 private static final String NOMBRE\_ARCHIVO = "Exportacion.xlsx";  
 private static final String[] tipos = {"Formato", "Norma", "Perfiles", "Crudos"};  
 private static final SimpleDateFormat formatoFecha = new SimpleDateFormat("dd/MMM/yyyy H:mm", Locale.ENGLISH);  
 private static final String GMT = "GMT-06:00";  
  
 private String nombrePlanta;  
 private String fechaInicio;  
 private String fechaFinal;  
 private BigDecimal planta;  
  
 private Driver driver = null;  
  
 private boolean inPlay = false;  
 @Autowired  
 private DAOCatPlanta daoCatPlanta;  
 @Autowired  
 private DAOVariable daoVariable;  
 @Autowired  
 private DaoNegocioTxCincominutalVar daoNegocioTxCincominutalVar;  
 @Autowired  
 private DAOCatEstatusVariable daoCatEstatusVariable;  
 @Autowired  
 private DaoNegocioTxPerfilesVariable daoNegocioTxPerfilesVariable;  
 @Autowired  
 private DAOCatEstatusPerfil daoCatEstatusPerfil;  
 @Autowired  
 private DAOCatTags daoCatTags;  
 @Autowired  
 @Qualifier("simpleAsyncTaskExecutor")  
 private AsyncTaskExecutor executor;  
  
 @Autowired  
 private ApplicationContext applicationContext;  
  
 @Autowired  
 BusinessCorreccionNorma businessCorreccionNorma;  
  
 @Value("${spring.datasource.url}")  
 private String dataSourceUrl;  
 @Value("${spring.datasource.username}")  
 private String dataSourceUser;  
 @Value("${spring.datasource.password}")  
 private String dataSourcePass;  
 @Value("${spring.datasource.driver.class}")  
 private String getDataSourceDriver;  
  
 @Value("${utils.exportadorexcel.path}")  
 private String path;  
  
  
 @Override  
 public DominioDyCNorma detectaErrores(DominioDyCNorma dom) throws DyCNormaException {  
 if (inPlay) {  
 System.out.println("La deteccion y correccion ya esta ejecutandose");  
 return dom;  
 }  
 inPlay = true;  
 CatPlanta planta = daoCatPlanta.findById(dom.getCatPlanta().getPlantaId()).orElse(new CatPlanta());  
 List<CatVariable> variables = daoVariable.findAllByTipoAndCatPlanta("A", planta);  
 System.out.printf("Variables A encontradas: %d\n", variables.size());  
 List<CatVariable> variablesCromatograficas = daoVariable.findAllByTipoAndCatPlanta("C", planta);  
 variables.addAll(variablesCromatograficas);  
 System.out.printf("Variables C encontradas: %d\n", variablesCromatograficas.size());  
 TaskSaveDeteccion taskSaveDeteccion = applicationContext.getBean(TaskSaveDeteccion.class);  
 List<Future<?>> futureList = new ArrayList<>();  
  
 for (CatVariable varCat : variables) {  
 taskSaveDeteccion.setCatVariable(varCat);  
 taskSaveDeteccion.run();  
// Future<?> future = executor.submit(taskSaveDeteccion);  
// futureList.add(future);  
 }  
  
 /\*  
 for (Future<?> future : futureList) {  
 while (!future.isDone()) {  
 try {  
 Thread.sleep(300);  
 } catch (InterruptedException e) {  
 }  
 }  
 }\*/  
  
 System.out.println("Todos los hilos TaskSaveDeteccion terminaron su proceso.");  
  
 dom = businessCorreccionNorma.corrigeErrores(dom);  
 System.out.println("Habilitando la funcionalidad ");  
 inPlay = false;  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 }  
  
 /\*\*  
 \* Usa la Clase <code>GeneradorExcel</code> para exportar y guardar en un cuaderno de Excel los  
 \* <code>TxCincominutal</code> contenidos dentro de los <code>TxCincominutalVar</code>  
 \*  
 \* @param dom bean que contiene los datos del rango de fechas  
 \* @param sTipo  
 \*/  
 @Override  
 public void generaExcel(DominioDyCNorma dom, String sTipo) throws DyCNormaException {  
  
 //pasar fechas de String a Date y generar la lista de TxCincominutalVar y generar el Excel  
 establecerFechasYPlanta(dom);  
  
  
 try {  
 Date[] rangosFechas = getDates();  
  
 System.out.printf("Tipo de Excel a generar: %s\n", sTipo);  
 List<TxPerfilesVariable> listaTxPerfilesVariables;  
 List<CatEstatusPerfil> listaCatEstatusPerfil;  
 List<CatTags> listaTags;  
  
 if (tipos[0].equals(sTipo)) { //Si el sTipo es igual a "Formato"  
 List<TxCincominutalVar> listaTxCincominutalVars = leerCincominutalVars(rangosFechas);  
 List<CatEstatusVariable> listaCatEstatusVariable = leerEstatusVariables();  
 listaTags = daoCatTags.getCatTagsByPlantaId(planta);  
  
 ExportadorExcel exportadorExcel = new ExportadorExcel(tipos[0], nombrePlanta, fechaInicio, fechaFinal);  
  
 if (!exportadorExcel.generarExcel(listaTxCincominutalVars, listaCatEstatusVariable, listaTags,  
 Globales.PATH\_DYC\_EX + File.separator + NOMBRE\_ARCHIVO\_F, "Formato")) {  
 System.out.println("No se pudo generar " + NOMBRE\_ARCHIVO\_F);  
 } else {  
 }  
 } else if (tipos[1].equals(sTipo)) { //Si el sTipo es igual a "Norma"  
 List<TxCincominutalVar> listaTxCincominutalVars = leerCincominutalVars(rangosFechas);  
 List<CatEstatusVariable> listaCatEstatusVariable = leerEstatusVariables();  
 listaTags = daoCatTags.getCatTagsByPlantaId(planta);  
  
 ExportadorExcel exportadorExcel = new ExportadorExcel(tipos[1], nombrePlanta, fechaInicio, fechaFinal);  
  
 if (!exportadorExcel.generarExcel(listaTxCincominutalVars, listaCatEstatusVariable, listaTags,  
 Globales.PATH\_DYC\_EX + File.separator + NOMBRE\_ARCHIVO\_N, "Norma")) {  
 System.out.println("No se pudo generar " + NOMBRE\_ARCHIVO\_N);  
 }  
 } else if (tipos[2].equals(sTipo)) { //Si el sTipo es igual a "Perfiles"  
 System.out.println("Recuperando registros. . .");  
 listaTxPerfilesVariables = daoNegocioTxPerfilesVariable.getTxPerfilesVariableByRango(rangosFechas[0],  
 rangosFechas[1], planta);  
 System.out.printf("Registros encontrados: %d \n", listaTxPerfilesVariables.size());  
 listaCatEstatusPerfil = daoCatEstatusPerfil.getCatEstatusPerfilByEstatusPerfilId();  
  
 ExportadorExcel exportadorExcel = new ExportadorExcel(tipos[2], nombrePlanta, fechaInicio, fechaFinal);  
 System.out.printf("Se encontraron %d estatus posibles. . .", listaCatEstatusPerfil.size());  
 if (!exportadorExcel.generarExcelPerfiles(listaTxPerfilesVariables, listaCatEstatusPerfil,  
 Globales.PATH\_DYC\_EX + File.separator + NOMBRE\_ARCHIVO\_PER, dom.getFechaFin())) {  
 System.out.println("No se pudo generar " + NOMBRE\_ARCHIVO\_PER);  
 }  
 } else { //Si sTipo no corresponde a ningun tipo definido  
 System.out.println("No se encuentra el tipo de Excel a generar en BusinessDeteccionNorma.generaExcel()");  
 }  
  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 System.out.println("Recuperando Lista de la Base de Datos...");  
 }  
  
 public void generaExcel(DominioDyCNorma dom) throws DyCNormaException {  
 establecerFechasYPlanta(dom);  
  
 try {  
 Date[] rangosFechas = getDates();  
  
 List<TxPerfilesVariable> listaTxPerfilesVariables = null;  
 List<CatEstatusPerfil> listaCatEstatusPerfil = null;  
 List<CatTags> listaTags = null;  
 System.out.printf("Generando documento: %s\n", Arrays.toString(dom.getTipoExportacion()));  
 List<TxCincominutalVar> listaTxCincominutalVars = null;  
 List<CatEstatusVariable> listaCatEstatusVariable = null;  
  
 ExportadorExcel exportadorExcel = null;  
  
 String path = this.path + File.separator + NOMBRE\_ARCHIVO;  
 System.out.println("ubicacion: " + path);  
  
 boolean bExportarPerfiles = false;  
 boolean bExportarFormatoONorma = false;  
  
 for (String tipo :  
 dom.getTipoExportacion()) {  
 if (tipo.equals(tipos[0]) || tipo.equals(tipos[1])) {  
 bExportarFormatoONorma = true;  
 exportadorExcel = new ExportadorExcel("Deteccion y Correccion", nombrePlanta,  
 fechaInicio, fechaFinal, dom.getTipoExportacion());  
 listaTxCincominutalVars = leerCincominutalVars(rangosFechas);  
 listaTags = daoCatTags.getCatTagsByPlantaId(planta);  
 listaCatEstatusVariable = leerEstatusVariables();  
 }  
 if (tipo.equals(tipos[2])) {  
 bExportarPerfiles = true;  
 exportadorExcel = new ExportadorExcel("Deteccion y Correccion", nombrePlanta,  
 fechaInicio, fechaFinal, dom.getTipoExportacion());  
 listaTxPerfilesVariables = daoNegocioTxPerfilesVariable.getTxPerfilesVariableByRango(rangosFechas[0],  
 rangosFechas[1], planta);  
 System.out.printf("Perfiles encontrados: %d \n", listaTxPerfilesVariables.size());  
 listaCatEstatusPerfil = daoCatEstatusPerfil.getCatEstatusPerfilByEstatusPerfilId();  
 }  
 if (tipo.equals(tipos[3])) {  
 exportadorExcel = new ExportadorExcel("Datos Crudos", nombrePlanta,  
 fechaInicio, fechaFinal, dom.getTipoExportacion());  
 }  
 }  
  
 if (!bExportarPerfiles && bExportarFormatoONorma && exportadorExcel.generarExcel(listaTxCincominutalVars, listaCatEstatusVariable, listaTags,  
 path, dom.getTipoExportacion(), dom.getFechaFin())) {  
 System.out.println("Excel generado correctamente");  
 } else if (bExportarPerfiles) {  
  
 if (!bExportarFormatoONorma && exportadorExcel.generarExcelPerfiles(listaTxPerfilesVariables,  
 listaCatEstatusPerfil, path, dom.getFechaFin())) {  
 System.out.println("Excel generado correctamente");  
 } else if (bExportarFormatoONorma && exportadorExcel.generarExcel(listaTxCincominutalVars,  
 listaCatEstatusVariable, listaTags, dom.getTipoExportacion(),  
 listaTxPerfilesVariables, listaCatEstatusPerfil, path, planta, dom.getFechaFin())) {  
 System.out.println("Excel generado correctamente");  
 }  
  
 }  
  
  
 //exportacion de datos crudos  
 if (dom.getTipoExportacion()[0].equals(tipos[3])) {  
 String tabla;  
 if (planta.equals(BigDecimal.ONE)){  
 tabla = "tx\_crudos";  
 } else {  
 tabla = "tx\_crudos\_sol";  
 }  
 //Leemos los meta-datos de los datos crudos  
 Connection connection = getConnection();  
 Statement statement = connection.createStatement();  
 SimpleDateFormat dia = new SimpleDateFormat("dd/MM/yyyy");  
 SimpleDateFormat minuto = new SimpleDateFormat("HH:mm:ss");  
 String sDiaInicio = dia.format(formatoFecha.parse(dom.getFechaInicio().trim()));  
 String sMinutoInicio = minuto.format(formatoFecha.parse(dom.getFechaInicio().trim()));  
 String sDiaFin = dia.format(formatoFecha.parse(dom.getFechaFin().trim()));  
 String sMinutoFin = minuto.format(formatoFecha.parse(dom.getFechaFin().trim()));  
// statement.executeUpdate("alter session set nls\_date\_format = 'DD/MM/YYYY HH24:MI:SS'") ;  
 /\*query += "SELECT \* FROM " + tabla  
 + " WHERE " + tabla + ".dia IS NOT NULL AND "  
 + tabla + ".dia = '" + sDia + "'";\*/  
 String query = "select \* from tx\_crudos where dia is not null " +  
 "and fecha between '" + sDiaInicio + " " + sMinutoInicio + "' " +  
 "and '" + sDiaFin + " " + sMinutoFin + "' " +  
 "order by to\_date(dia || ' ' || minuto,'DD/MM/YYYY HH24:MI:SS') asc";  
 System.out.println(query);  
 ResultSet resultSet = statement.executeQuery(query);  
 System.out.printf("Columnas encontradas: %d\n", resultSet.getMetaData().getColumnCount());  
 exportadorExcel.generarExcel(resultSet,path);  
  
 connection.close();  
  
 //TODO: exportacion de datos crudos  
 }  
  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 }  
  
 private void establecerFechasYPlanta(DominioDyCNorma dom) {  
 fechaInicio = dom.getFechaInicio();  
 fechaFinal = dom.getFechaFin();  
 planta = dom.getCatPlanta().getPlantaId();  
  
  
 nombrePlanta = (planta.equals(BigDecimal.ONE)) ? "ELECTRICIDAD AGUILA DE TUXPAN" : " ELECTRICIDAD SOL DE TUXPAN";  
 }  
  
 private Date[] getDates() throws ParseException {  
 System.out.println("FechaInicio: " + fechaInicio);  
 Date[] rangosFechas = new Date[2];  
 rangosFechas[0] = formatoFecha.parse(fechaInicio.trim());  
 System.out.println("FechaFinal: " + fechaFinal);  
 rangosFechas[1] = formatoFecha.parse(fechaFinal.trim());  
 System.out.print("Fecha inicial: ");  
 System.out.println(rangosFechas[0]);  
 System.out.print("Fecha final: ");  
 System.out.println(rangosFechas[1]);  
 return rangosFechas;  
 }  
  
 private List<TxCincominutalVar> leerCincominutalVars(Date[] rangosFechas) {  
 List<TxCincominutalVar> listaTxCincominutalVars;  
 System.out.println("Recuperando registros. . .");  
 listaTxCincominutalVars = daoNegocioTxCincominutalVar.getTxCincominutalVarByRango(rangosFechas[0],  
 rangosFechas[1], planta);  
 System.out.printf("Registros encontrados: %d \n", listaTxCincominutalVars.size());  
 return listaTxCincominutalVars;  
 }  
  
 private List<CatEstatusVariable> leerEstatusVariables() {  
 List<CatEstatusVariable> listaCatEstatusVariable;  
 listaCatEstatusVariable = daoCatEstatusVariable.getCatEstatusVariableByEstatusVariableId();  
 System.out.printf("Se encontraron %d estatus posibles. . .", listaCatEstatusVariable.size());  
 return listaCatEstatusVariable;  
 }  
  
 private synchronized Connection getConnection() throws SQLException {  
 if (driver == null) {  
 //Carga el driver JDBC  
 try {  
 Class jdbcDriverClass = Class.forName(getDataSourceDriver);  
 driver = (Driver) jdbcDriverClass.newInstance();  
 DriverManager.registerDriver(driver);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 return DriverManager.getConnection(dataSourceUrl, dataSourceUser, dataSourcePass);  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\dyc\BusinessDeteccionNormaPerfiles.java

package mx.mbi.dgc.business.dyc;  
  
import java.math.BigDecimal;  
  
import java.util.ArrayList;  
import java.util.Hashtable;  
import java.util.List;  
import java.util.Map;  
import java.util.Set;  
  
import mx.mbi.dgc.dao.DAOCatEstatusPerfil;  
import mx.mbi.dgc.dao.DAOCatPlanta;  
import mx.mbi.dgc.dao.DAOTxPerfilesVariable;  
import mx.mbi.dgc.dao.negocio.DAOVariablePerfil;  
import mx.mbi.dgc.dao.negocio.DaoNegocioCatMedidores;  
import mx.mbi.dgc.dao.negocio.DaoNegocioTxPerfilesVariable;  
import mx.mbi.dgc.dominio.DominioDyCNorma;  
import mx.mbi.dgc.dominio.Evaluacion;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatEstatusPerfil;  
import mx.mbi.dgc.jpa.entity.CatMedidores;  
import mx.mbi.dgc.jpa.entity.CatMesDiaFirma;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.jpa.entity.CatRangoVariable;  
import mx.mbi.dgc.jpa.entity.CatVariable;  
import mx.mbi.dgc.jpa.entity.CatVariablePerfil;  
import mx.mbi.dgc.jpa.entity.TxPerfilesVariable;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.joda.time.DateTime;  
import org.joda.time.format.DateTimeFormat;  
import org.joda.time.format.DateTimeFormatter;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessDeteccionNormaPerfiles implements IfaceDeteccionNormaPerfiles {  
 private static final Logger LOG = LogManager.getLogger(BusinessDeteccionNorma.class);  
   
 @Autowired  
 private DAOVariablePerfil daoCatVariablePerfil;  
   
 @Autowired  
 private DAOCatPlanta daoCatPlanta;  
 @Autowired  
 private DAOCatEstatusPerfil daoCatEstatusPerfil;  
   
 @Autowired  
 private DaoNegocioTxPerfilesVariable daoNegocioTxPerfilesVariable;  
   
 @Autowired  
 private DAOTxPerfilesVariable daoTxPerfilesVariable;   
   
 @Autowired  
 private DaoNegocioCatMedidores daoNegocioCatMedidores;  
   
 CatPlanta catPlanta;  
 TxPerfilesVariable varUltimaValida;  
 Evaluacion evalExitosa;  
 CatEstatusPerfil catEstatusVariable;  
  
 @Override  
 public DominioDyCNorma detectaErrores(DominioDyCNorma dom) throws DyCNormaException {  
 BigDecimal activo = new BigDecimal("1");  
 evalExitosa = new Evaluacion();  
 evalExitosa.setRegla(new BigDecimal(2)); //2 = válido  
 evalExitosa.setStatus(true);  
 List <Map> tuplaPersistencia = new ArrayList<Map>();  
 Map<String, Object> mapPersistencia = new Hashtable<>();  
 catPlanta = daoCatPlanta.findById(dom.getCatPlanta().getPlantaId()).orElse(new CatPlanta());  
 catEstatusVariable = daoCatEstatusPerfil.findById(new BigDecimal(2)).orElse(new CatEstatusPerfil());//2 = VALIDO  
 try {  
 for (CatMedidores catMedidor : daoNegocioCatMedidores.findAllByCatPlanta(catPlanta)) {  
 for (CatVariablePerfil catVariable : daoCatVariablePerfil.findAllByCatPlanta(catPlanta)) {  
 List<TxPerfilesVariable> listaTxPerfilesVariable = null;  
 listaTxPerfilesVariable = daoNegocioTxPerfilesVariable.getAllTxPerfilesVariable(  
 catVariable.getVariablePerfilId(),   
 dom.getCatPlanta().getPlantaId(),  
 catMedidor.getMedidorId());  
 if(listaTxPerfilesVariable != null && listaTxPerfilesVariable.size() > 0) {  
 varUltimaValida = listaTxPerfilesVariable.get(0);  
 } else {  
 System.out.println("No hay datos, o existe inconsistencia de datos: varUltimaValida es null. . .");  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 }  
 if(tuplaPersistencia != null && tuplaPersistencia.size() > 0){  
 varUltimaValida = persisteTuplaPersistencia(tuplaPersistencia);  
 }  
   
 tuplaPersistencia = new ArrayList<Map>();  
 mapPersistencia = new Hashtable<>();  
   
 mapPersistencia.put("Evaluacion", evalExitosa);  
 mapPersistencia.put("TxPerfilesVariable", varUltimaValida);  
 tuplaPersistencia.add(mapPersistencia);  
   
 TxPerfilesVariable var = null;  
 int errores = 0;  
 for (TxPerfilesVariable var\_ : listaTxPerfilesVariable) {  
 var = var\_;  
 Evaluacion eval = null;  
   
 if (var.getTxCincominutal()  
 .getCatMesDiaFirma()  
 .getCatFirmaCincominutal()  
 .getFirma()  
 .equals("00000005") && isPrimeraDiaDelMes(var.getTxCincominutal().getCatMesDiaFirma())) {  
 eval = validaIPM(var);  
 } else {  
 if (var.getTxCincominutal()  
 .getCatMesDiaFirma()  
 .getCatFirmaCincominutal()  
 .getFirma()  
 .equals("23550000") && isUltimoDiaDelMes(var.getTxCincominutal().getCatMesDiaFirma())) {  
 eval = validaUPM(var, varUltimaValida);  
 } else {  
 eval = validaReglas(var, varUltimaValida);  
 }  
 }  
   
 mapPersistencia = new Hashtable<>();  
 mapPersistencia.put("Evaluacion", eval);  
 mapPersistencia.put("TxPerfilesVariable", var);  
 tuplaPersistencia.add(mapPersistencia);  
 System.out.println("eval.isStatus() : " + eval.isStatus());  
 if (eval.isStatus()) {  
 System.out.println("tuplaPersistencia.size(): " + tuplaPersistencia.size());  
 varUltimaValida = persisteTuplaPersistencia(tuplaPersistencia);  
   
 tuplaPersistencia = new ArrayList<Map>();  
 mapPersistencia = new Hashtable<>();  
 mapPersistencia.put("Evaluacion", evalExitosa);  
 mapPersistencia.put("TxPerfilesVariable", varUltimaValida);  
 tuplaPersistencia.add(mapPersistencia);  
 } else {  
 errores++;  
 System.out.println("errores: " + errores);  
 }  
 }  
 }  
 }  
   
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en BusinessDeteccionNorma.detectaErrores: " + ex);  
 }  
 }  
   
   
 private TxPerfilesVariable persisteTuplaPersistencia( List <Map> tuplaPersistencia){  
 TxPerfilesVariable nueva = null;  
 Map mapVar = tuplaPersistencia.get(tuplaPersistencia.size() - 1);  
  
 TxPerfilesVariable retorno = (TxPerfilesVariable)mapVar.get("TxPerfilesVariable");  
 tuplaPersistencia.remove(tuplaPersistencia.size() - 1);  
 for(int i = 0; i <tuplaPersistencia.size(); i++){  
 mapVar = tuplaPersistencia.get(i);  
 CatEstatusPerfil status = daoCatEstatusPerfil.findById(((Evaluacion)mapVar.get("Evaluacion")).getRegla()).orElse(new CatEstatusPerfil());  
 TxPerfilesVariable txCVar = ((TxPerfilesVariable)mapVar.get("TxPerfilesVariable"));  
 nueva = new TxPerfilesVariable();  
 nueva.setCatMedidores(txCVar.getCatMedidores());  
 nueva.setCatVariablePerfil(txCVar.getCatVariablePerfil());  
 nueva.setPrincipal(new BigDecimal(1));  
 txCVar.setPrincipal(new BigDecimal(0));  
 nueva.setTxCincominutal(txCVar.getTxCincominutal());  
 nueva.setValor(txCVar.getValor());  
 nueva.setCatEstatusPerfil(status);  
 daoTxPerfilesVariable.save(txCVar);  
 daoTxPerfilesVariable.save(nueva);  
 }  
   
 return retorno;  
 }  
  
 private CatRangoVariable getRangoVariable(Set<CatRangoVariable> rangos, CatVariable var, CatPlanta planta) {  
 for (CatRangoVariable ran : rangos) {  
 if (ran.getCatPlanta().getPlantaId() == planta.getPlantaId() && var.getVariableId().intValue() == ran.getCatVariable().getVariableId().intValue()) {  
 return ran;  
 }  
 }  
 System.out.println("################ CatRangoVariable nulo. . ."+var.getVariableId());  
 return null;  
 }  
   
 private Evaluacion validaUPM(TxPerfilesVariable txCvar, TxPerfilesVariable varPrimeraValida) {  
 double nuevoValor = txCvar.getValor().doubleValue();  
 double valor = varPrimeraValida.getValor().doubleValue() - txCvar.getValor().doubleValue();  
 Evaluacion eval = new Evaluacion();  
 eval.setRegla(new BigDecimal(2)); //2 = válido  
 eval.setStatus(true);  
 //Validar congelamiento = 15  
 boolean quitarAddon = false;  
 if (valor == 0) {  
 nuevoValor = txCvar.getValor().doubleValue() + 0.01;   
 eval.setStatus(true);  
 eval.setRegla(new BigDecimal(28)); //28 CONGELADA ULTIMA DEL MES  
 }  
  
 Evaluacion evalLimites = validaReglaLimites(txCvar);  
 if(evalLimites != null){  
 nuevoValor = txCvar.getValor().doubleValue();  
 return evalLimites;  
 }  
 txCvar.setValor(new BigDecimal(nuevoValor));  
 return eval;  
 }  
  
 private Evaluacion validaIPM(TxPerfilesVariable txCvar) {  
 //Valida limites inferior y superior  
 Evaluacion eval = new Evaluacion();  
 eval.setRegla(new BigDecimal(2)); //2 VALIDO  
 eval.setStatus(true);  
 Evaluacion evalLimites = validaReglaLimites(txCvar);  
 if(evalLimites != null){  
 return evalLimites;  
 }  
 return eval;  
 }  
   
 private boolean isUltimoDiaDelMes(CatMesDiaFirma inicioTemporal){  
 boolean retorno = false;  
 String pattern = "yyyy\_MMdd";  
 String cadenaTiempo = inicioTemporal.getCatMesDia().getCatAnioMes().getAnioMes() + inicioTemporal.getCatMesDia().getDia();  
 DateTimeFormatter builderDateTimeFormat = DateTimeFormat.forPattern(pattern);  
 DateTime inicioTemporalDateTime = DateTime.parse(cadenaTiempo, builderDateTimeFormat);  
 int dia = inicioTemporalDateTime.getDayOfMonth();  
 int finMes = inicioTemporalDateTime.dayOfMonth().getMaximumValue();  
 if(dia == finMes){  
 retorno = true;  
 }  
 return retorno;  
 }  
   
 private boolean isPrimeraDiaDelMes(CatMesDiaFirma inicioTemporal){  
 boolean retorno = false;  
 String pattern = "yyyy\_MMdd";  
 String cadenaTiempo =inicioTemporal.getCatMesDia().getCatAnioMes().getAnioMes() + inicioTemporal.getCatMesDia().getDia();  
 DateTimeFormatter builderDateTimeFormat = DateTimeFormat.forPattern(pattern);  
 DateTime inicioTemporalDateTime = DateTime.parse(cadenaTiempo, builderDateTimeFormat);  
 int dia = inicioTemporalDateTime.getDayOfMonth();  
 if(dia == 1){  
 retorno = true;  
 }  
 return retorno;  
 }  
   
 private Evaluacion validaReglas(TxPerfilesVariable varValidar, TxPerfilesVariable varPrimeraValida) {  
 double valor = varPrimeraValida.getValor().doubleValue() - varValidar.getValor().doubleValue();  
 Evaluacion eval = new Evaluacion();  
 eval.setRegla(new BigDecimal(2)); //2 = válido  
 eval.setStatus(true);  
 //Validar congelamiento = 21  
 if (valor == 0) {  
 eval.setStatus(false);  
 eval.setRegla(new BigDecimal(21));  
 return eval;  
 }  
   
 Evaluacion evalLimites = validaReglaLimites(varValidar);  
 if(evalLimites != null){  
 return evalLimites;  
 }  
 return eval;  
 }  
   
 private Evaluacion validaReglaLimites(TxPerfilesVariable txCvar){  
 Evaluacion eval = null;  
 //22 INVALIDO POR FUERA LIMITE INFERIOR  
 if (txCvar.getValor().doubleValue() < txCvar.getCatVariablePerfil()  
 .getLimiteInferior()  
 .doubleValue()) {  
 eval = new Evaluacion();  
 eval.setStatus(false);  
 eval.setRegla(new BigDecimal(22));  
 }  
 //23 INVALIDO POR FUERA LIMITE SUPERIOR  
 if (txCvar.getValor().doubleValue() > txCvar.getCatVariablePerfil()  
 .getLimiteSuperior()  
 .doubleValue()) {  
 eval.setStatus(false);  
 eval.setRegla(new BigDecimal(23));  
 }  
 return eval;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\dyc\BusinessDyCNorma.java

package mx.mbi.dgc.business.dyc;  
  
import mx.mbi.dgc.dominio.DominioDyCNorma;  
import mx.mbi.dgc.dominio.DominioReporteDyC;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessDyCNorma implements IfaceDyCNorma {  
  
  
 @Override  
 public DominioReporteDyC getTags(DominioDyCNorma dom) throws DyCNormaException {  
 // TODO Implement this method  
 return null;  
 }  
  
 @Override  
 public DominioReporteDyC getValoresTags(DominioReporteDyC tags) throws DyCNormaException {  
 // TODO Implement this method  
 return null;  
 }  
   
   
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\dyc\IfaceCorreccionNorma.java

package mx.mbi.dgc.business.dyc;  
  
import mx.mbi.dgc.dominio.DominioDyCNorma;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
  
public interface IfaceCorreccionNorma {  
 DominioDyCNorma corrigeErrores(DominioDyCNorma dom) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\dyc\IfaceCorreccionNormaPerfiles.java

package mx.mbi.dgc.business.dyc;  
  
import mx.mbi.dgc.dominio.DominioDyCNorma;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
  
public interface IfaceCorreccionNormaPerfiles {  
 DominioDyCNorma corrigeErrores(DominioDyCNorma dom) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\dyc\IfaceDeteccionNorma.java

package mx.mbi.dgc.business.dyc;  
  
import mx.mbi.dgc.dominio.DominioDyCNorma;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
  
public interface IfaceDeteccionNorma {  
 public DominioDyCNorma detectaErrores(DominioDyCNorma dom) throws DyCNormaException;  
  
 /\*\*  
 \* @param dom Contiene los datos de rangos de fecha y planta para poder realizar la exportacion a Excel.  
 \* @param sTipo El tipo de Excel a exportar.  
 \* @return  
 \* @throws DyCNormaException  
 \*/  
 public void generaExcel(DominioDyCNorma dom, String sTipo) throws DyCNormaException;  
  
 public void generaExcel(DominioDyCNorma dom) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\dyc\IfaceDeteccionNormaPerfiles.java

package mx.mbi.dgc.business.dyc;  
  
import mx.mbi.dgc.dominio.DominioDyCNorma;  
import mx.mbi.dgc.excepcion.DyCNormaException;   
  
public interface IfaceDeteccionNormaPerfiles {  
 DominioDyCNorma detectaErrores(DominioDyCNorma dom) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\dyc\IfaceDyCNorma.java

package mx.mbi.dgc.business.dyc;  
  
import mx.mbi.dgc.dominio.DominioDyCNorma;  
import mx.mbi.dgc.dominio.DominioReporteDyC;  
import mx.mbi.dgc.dominio.DominioTags;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
  
public interface IfaceDyCNorma {  
 DominioReporteDyC getTags(DominioDyCNorma dom) throws DyCNormaException;  
 DominioReporteDyC getValoresTags(DominioReporteDyC tags) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\dyc\TaskCorreccionAmbientales.java

package mx.mbi.dgc.business.dyc;  
  
import java.math.BigDecimal;  
  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatPlanta;  
import mx.mbi.dgc.dao.negocio.DAOCincoVar;  
import mx.mbi.dgc.jpa.entity.\*;  
import mx.mbi.dgc.utils.DominioTool;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.beans.factory.annotation.Qualifier;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.annotation.Scope;  
import org.springframework.core.task.AsyncTaskExecutor;  
import org.springframework.stereotype.Component;  
  
@Component  
@Scope("prototype")  
public class TaskCorreccionAmbientales implements Runnable {  
 private class EstatusNormaVariable{  
 BigDecimal estatusDn = null;  
 BigDecimal estatusCn = null;  
 }  
  
 private enum Direcciones {  
 ANTERIOR,  
 SIGUIENTE;  
 }  
  
  
 private static final Logger LOG = LogManager.getLogger(BusinessCorreccionNorma.class);  
  
 @Autowired  
 private DAOCincoVar daoCincoVar;  
  
 @Autowired  
 private DAOCatPlanta daoCatPlanta;  
  
 @Autowired  
 private DominioTool tool;  
  
 @Autowired  
 @Qualifier("simpleAsyncTaskExecutor")  
 private AsyncTaskExecutor executor;  
  
 @Autowired  
 private ApplicationContext applicationContext;  
  
  
 private CatPlanta catPlanta;  
 private CatVariable catVariable;  
  
 private static BigDecimal noCorregido = new BigDecimal("-1"); //-1 = NO CORREGIDO  
 private static BigDecimal valido = new BigDecimal("27"); //28 = VALIDO  
 private static BigDecimal validoSinCorregir = new BigDecimal("28"); //28 = VALIDO SIN CORREGIR  
  
 private Integer delay;  
  
 public TaskCorreccionAmbientales() {  
 super();  
 }  
  
 @Override  
 public void run() {  
 System.out.println("Corriendo proceso de Deteccion de Norma");  
 this.catPlanta = daoCatPlanta.findById(catVariable.getCatPlanta().getPlantaId()).orElse(new CatPlanta());  
  
 try {  
 //primer cincominutal de estatusDn = 27 y estatusCn = 28  
 TxCincominutalVar varPrimeraValida = getMinVarPrimeraValida(catVariable);  
  
 if (varPrimeraValida != null) {  
 System.out.println("Primera valida encontrada, buscando pivotes");  
  
  
 /\*Hasta aqui, para una variable dada, verifica si había un cincominutalVar de {estatusDn = 27}  
 y {estatusCn = 28}, para la planta definida en catPlanta (sic.), y si existe, toma todos los  
 cincominutalVars de esa planta y esa variable que tengan {estatusDn = 1} y los vuelve a guardar  
 en la base de datos con {EstatusDn = 27} y con {fechaUm = new Date()} y {FechaDn = FechaUm}\*/  
 List<TxCincominutalVar> listaTxCincoVarNoCorregidos;  
  
 // Recupera la lista de cincominutalVars de {estatusCn = -1} y de cincominutal estrictamente posterior  
 // al primer cincominutalVar de {estatusDn = 27} y {estatusCn = 28}, de la misma planta  
 // y la misma variable  
 listaTxCincoVarNoCorregidos =  
 daoCincoVar.getListaAmbientales(noCorregido,  
 getCatVariable().getVariableId(),  
 varPrimeraValida.getTxCincominutal().getCincominutalId(),  
 catPlanta.getPlantaId());  
 System.out.printf("Se encontraron %d registros sin corregir para la variable %s\n",  
 listaTxCincoVarNoCorregidos.size(), catVariable.getNombre());  
  
 //corregirALaRicardo(varPrimeraValida, listaTxCincoVarNoCorregidos);  
  
 TxCincominutalVar buffer = null;  
 ArrayList<TxCincominutalVar> tuplaACorregir = new ArrayList<>();  
  
 //Los cincominutales se corrigen según el número de valores erróneos consecutivos  
 for (TxCincominutalVar cincominutal : listaTxCincoVarNoCorregidos) {  
 System.out.println("Corrigiendo...");  
 if (buffer == null) {  
 //Esto ocurre al principio de cada tupla  
 tuplaACorregir.add(cincominutal);  
 } else if (sonConsecutivos(buffer, cincominutal)) {  
 tuplaACorregir.add(cincominutal);  
 } else {  
 System.out.printf("Corrigiendo %d cincominutales consecutivos\n", tuplaACorregir.size());  
 corregir(tuplaACorregir);  
 //Crea una nueva tupla a corregir, limpia el buffer.  
 tuplaACorregir = new ArrayList<>();  
 buffer = null;  
 continue;  
 }  
 buffer = cincominutal;  
 }  
 if (tuplaACorregir.size() > 0) {  
 corregir(tuplaACorregir);  
 }  
 } else {  
 System.out.println("No Existen datos para realizar la correccion en TaskCorreccionAmbientales");  
 }  
  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 }  
  
 @SuppressWarnings("Duplicates")  
 private void corregirALaAntigua(TxCincominutalVar varPrimeraValida, List<TxCincominutalVar> listaTxCincoVarNoCorregidos) throws Exception {  
 System.out.printf("Tamano de la lista: %d\n", listaTxCincoVarNoCorregidos.size());  
 List<TxCincominutalVar> tuplaCorreccion = new ArrayList<>();  
 int variablesProcesadas = 0;  
 //si la lista existe y no esta vacia  
 if (listaTxCincoVarNoCorregidos != null && listaTxCincoVarNoCorregidos.size() > 0) {  
 //Se añade el primer cincominutal de estatusDn = 27 y estatusCn = 28 a tuplaCorreccion.  
 tuplaCorreccion.add(varPrimeraValida);  
 //Tiempo de inicio de proceso de corrección  
 Date contadorDeTiempoInicio = new Date();  
 //por cada elemento de la lista de cincominutalVars de {estatusCn = -1} posteriores a la primera válida  
 for (TxCincominutalVar var : listaTxCincoVarNoCorregidos) {  
 System.out.printf("Valor del cincominutal %d-%s: %f, estatus: %d\n",  
 var.getTxCincominutal().getCincominutalId().intValue(),  
 var.getCatVariable().getNombre(),  
 var.getValorFormato().floatValue(),  
 var.getEstatusDn().intValue());  
 //Para mostrar la velocidad de correccion de variables  
 if (variablesProcesadas>=1000){  
 Date contadorDeTiempoFinal = new Date();  
 Long contadorDeTiempo = contadorDeTiempoFinal.getTime()-contadorDeTiempoInicio.getTime();  
 System.out.printf("Tardo %d milisegundos en procesar " +  
 "los ultimos %d registros\n", contadorDeTiempo, variablesProcesadas);  
 float velocidad = variablesProcesadas/(contadorDeTiempo/1000f);  
 System.out.printf("Se procesan a %f variables por segundo\n", velocidad);  
 //vuelve a empezar la cuenta para medir la velocidad de las siguientes 1000 variables  
 variablesProcesadas = 0;  
 contadorDeTiempoInicio = contadorDeTiempoFinal;  
 }  
  
 //Se añade el elemento a tuplaCorreccion.  
 tuplaCorreccion.add(var);  
  
 //Si el estatus de Deteccion de Norma de algún elemento de la lista de no corregidos  
 // posteriores al primer cincominutalVar valido es "valido"  
 if (var.getEstatusDn().intValue() == valido.intValue()) { //es valida  
 System.out.println("Encontrado valor valido");  
 //Guarda el primer elemento de tuplaCorreccion (varPrimeraValida) en la BD  
 // tal que {valorNorma = valorFormato}, {estatusCn = validoSinCorregir (28)}  
 // {fechaUm = fecha actual} y {fechaCn = fechaUm}  
 actualizarPrimeraValida(tuplaCorreccion.get(0));  
  
 //(si es válido &&) si tuplaCorreccion contiene 2 elementos, lo cuál significa que ambos son válidos  
 if (tuplaCorreccion.size() == 2) {  
 //TODO: ¿cambiar el estatus a 28?  
 for (TxCincominutalVar elementoValido :  
 tuplaCorreccion) {  
 EstatusNormaVariable estatus = new EstatusNormaVariable();  
 estatus.estatusCn = validoSinCorregir;  
 actualizarVariable(elementoValido, estatus);  
 }  
 //Borra tuplaCorreccion  
 tuplaCorreccion = new ArrayList<>();  
 //Añade el elemento actual (que es válido) de la lista de no corregidos,  
 // al principio de tuplaCorreccion  
 tuplaCorreccion.add(var);  
 //Termina la iteración actual y pasa al elemento no corregido SIGUIENTE  
 variablesProcesadas++;  
 continue;  
 }  
  
 //Si hay un cincominutalVar con estatusDn "valido", aplica la corrección a tuplaCorreccion  
 // borra tuplaCorreccion  
 // y añade el cincominutalVar (que es válido)a la nueva tuplaCorreccion  
  
  
 List<TxCincominutalVar> corregidos = corregirAmbientales(tuplaCorreccion);  
  
 System.out.printf("Guardando %d valores\n", corregidos.size());  
 for (TxCincominutalVar cincominutalVar :  
 corregidos) {  
 System.out.printf("Guardando %f...\n", cincominutalVar.getValorNorma().floatValue());  
 }  
 daoCincoVar.saveAll(corregidos);  
 /\*  
 synchronized (daoCincoVar) {  
 daoCincoVar.saveAll(corregidos);  
 }\*/  
 tuplaCorreccion = new ArrayList<>();  
 tuplaCorreccion.add(var);  
 } else {  
 tuplaCorreccion.add(var);  
 }  
 variablesProcesadas++;  
 }  
  
 daoCincoVar.flush();  
 //Después de revisar todos los cincominutalVars no corregidos, guarda el primero en la base de datos  
 // con {estatusCn = validoSinCorregir} y fechaUm y fechaCm actuales  
 // tuplaCorreccion siempre tiene al menos un elemento así que siempre corre.  
 //en su commit del 14 de enero RTC cambió de >1 a >0  
 if(tuplaCorreccion.size() > 0){  
 actualizarPrimeraValida(tuplaCorreccion.get(0));  
 }  
 }  
 }  
  
 private Boolean sonConsecutivos(TxCincominutalVar primero, TxCincominutalVar segundo) {  
 BigDecimal mesDiaFirmaPrimero = primero.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId();  
 BigDecimal mesDiaFirmaSegundo = segundo.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId();  
  
 return mesDiaFirmaPrimero.equals(mesDiaFirmaSegundo.subtract(BigDecimal.ONE));  
 }  
  
 @SuppressWarnings("Duplicates")  
 private void corregir(ArrayList<TxCincominutalVar> tuplaACorregir) {  
 TxCincominutalVar primerValor = tuplaACorregir.get(0);  
 TxCincominutalVar ultimoValor = tuplaACorregir.get(tuplaACorregir.size() - 1);  
 TxCincominutalVar valorAnterior = getValorConsecutivo(primerValor, Direcciones.ANTERIOR);  
 TxCincominutalVar valorPosterior = getValorConsecutivo(ultimoValor, Direcciones.SIGUIENTE);  
  
 if (valorAnterior == null) {  
 valorAnterior = primerValor;  
 }  
 if (valorPosterior == null) {  
 valorPosterior = ultimoValor;  
 }  
 if (valorPosterior.getEstatusDn().equals(BigDecimal.ONE)) {  
 actualizarPivote(valorPosterior);  
 }  
  
 boolean bValorAnteriorEsValido = valorAnterior.getEstatusDn().equals(valido);  
 boolean bValorPosteriorEsValido = valorPosterior.getEstatusDn().equals(valido);  
 boolean bValorAnteriorEsPivote = valorAnterior.getEstatusDn().equals(BigDecimal.ONE);  
 boolean bValorPosteriorEsPivote = valorPosterior.getEstatusDn().equals(BigDecimal.ONE);  
 boolean bValorAnteriorEsIgnorado = valorAnterior.getEstatusDn().equals(new BigDecimal("-5"));  
 boolean bValorPosteriorEsIgnorado = valorPosterior.getEstatusDn().equals(new BigDecimal("-5"));  
  
 if ((!bValorAnteriorEsValido || !bValorPosteriorEsValido)) {  
 System.out.printf("Estatus primer valor: %d\nEstatus ultimo valor: %d\n",  
 valorAnterior.getEstatusDn().intValue(),  
 valorPosterior.getEstatusDn().intValue());  
 System.out.printf("No se puede corregir tupla de %s para el cincominutal por falta de datos validos %s\n",  
 catVariable.getNombre(), tuplaACorregir.get(0).getTxCincominutal().getCatMesDiaFirma().getFechaFirma().toString());  
  
 if (bValorAnteriorEsPivote) {  
 EstatusNormaVariable estatus = new EstatusNormaVariable();  
 estatus.estatusDn = valido;  
 estatus.estatusCn = validoSinCorregir;  
 actualizarVariable(valorAnterior, estatus);  
 tuplaACorregir.add(0, valorAnterior);  
 } else if (bValorAnteriorEsIgnorado) {  
 tuplaACorregir.remove(0);  
 }  
  
 if (bValorPosteriorEsPivote) {  
 EstatusNormaVariable estatus = new EstatusNormaVariable();  
 estatus.estatusDn = valido;  
 estatus.estatusCn = validoSinCorregir;  
 actualizarVariable(valorAnterior, estatus);  
 tuplaACorregir.add(0, valorAnterior);  
 } else if (bValorPosteriorEsIgnorado){  
 tuplaACorregir.remove(tuplaACorregir.size()-1);  
 }  
 corregir(tuplaACorregir);  
 return;  
 }  
  
 if (!valorAnterior.equals(primerValor)) {  
 tuplaACorregir.add(0, valorAnterior);  
 }  
  
 if (!valorPosterior.equals(ultimoValor)) {  
 tuplaACorregir.add(valorPosterior);  
 }  
  
  
 try {  
 corregirALaAntigua(valorAnterior,tuplaACorregir);  
 //System.out.printf("Guardando %d valores corregidos", corregidos.size());  
 //daoCincoVar.saveAll(corregidos);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 }  
  
 private TxCincominutalVar getValorConsecutivo(TxCincominutalVar valor, Direcciones direccion) {  
 BigDecimal mesDiaFirmaId = valor.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId();  
 BigDecimal variableId = valor.getCatVariable().getVariableId();  
 BigDecimal catPlanta = this.catPlanta.getPlantaId();  
 TxCincominutalVar valorConsecutivo;  
  
 try {  
 if (direccion == Direcciones.ANTERIOR) {  
 valorConsecutivo = daoCincoVar.getTxCincoVarAnterior(mesDiaFirmaId.subtract(BigDecimal.ONE), variableId, catPlanta);  
 } else {  
 valorConsecutivo = daoCincoVar.getTxCincoVarAnterior(mesDiaFirmaId.add(BigDecimal.ONE), variableId, catPlanta);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 valorConsecutivo = null;  
 }  
 return valorConsecutivo;  
 }  
  
 /\*\*  
 \* Version refactorizada (mas general) de actualizarPrimeraValida y ActualizarPivote  
 \* @param variable una cincominutalVar  
 \* @param estatus objeto que contiene los valores a cambiar, los valores que no se cambian deben dejarse como null.  
 \*/  
 @SuppressWarnings("Duplicates")  
 private void actualizarVariable(TxCincominutalVar variable, EstatusNormaVariable estatus){  
 variable.setValorNorma(variable.getValorFormato());  
 variable.setFechaUm(new Date());  
 if (estatus.estatusDn!=null){  
 variable.setEstatusDn(estatus.estatusDn);  
 variable.setFechaDn(variable.getFechaUm());  
 }  
 if (estatus.estatusCn!=null){  
 variable.setEstatusCn(estatus.estatusCn);  
 variable.setFechaCn(variable.getFechaUm());  
 }  
 daoCincoVar.save(variable);  
 }  
  
  
 /\*\*  
 \* Es lo mismo que actualizarPivote solo que pone el estatusCn en validoSinCorregir en lugar de valido.  
 \* Se pueden refactorizar a un solo método.  
 \* @param primeraValida un txCincominutalVar que se supone es el primer cincominutalVar de estatusDn = 27 y estatusCn = 28  
 \*/  
 private void actualizarPrimeraValida(TxCincominutalVar primeraValida){  
 //TODO: Refactorizar a actualizarCincominutalVar()  
 primeraValida.setValorNorma(primeraValida.getValorFormato());  
 primeraValida.setEstatusCn(validoSinCorregir);  
 primeraValida.setFechaUm(new Date());  
 primeraValida.setFechaCn(primeraValida.getFechaUm());  
 daoCincoVar.save(primeraValida);  
 /\*synchronized (daoCincoVar) {  
 daoCincoVar.save(primeraValida);  
 }\*/  
 }  
  
 /\*\*  
 \* Cambia el estatusDn de un cincominutal (pivote) y sus fechaUm y fechaDn y lo guarda en la BD.  
 \* Es lo mismo que actualizarPrimeraValida y se pueden refactorizar ver {@link #actualizarPrimeraValida(TxCincominutalVar) actualizarPrimeraValida}.  
 \* @param pivote un cincominutal que se supone es de {EstatusDn = 1}  
 \* @return el mismo "pivote" con {EstatusDn = 27} y con {fechaUm = new Date()} y {FechaDn = FechaUm}. Realmente nunca se usa en el codigo actual.  
 \*/  
 private TxCincominutalVar actualizarPivote(TxCincominutalVar pivote){  
 pivote.setValorNorma(pivote.getValorFormato());  
 pivote.setEstatusDn(valido); //los pivotes tienen (estatuDn = 1), los cambia a 27  
 pivote.setEstatusCn(validoSinCorregir);  
 pivote.setFechaUm(new Date()); //se registra la fecha actual en fechaUm  
 pivote.setFechaDn(pivote.getFechaUm()); //como se acaba de definir fechaUm, realmente fechaDn = fechaUm que es la fecha actual  
  
 System.out.println("Pre-save");  
 daoCincoVar.save(pivote);  
 System.out.println("Post-save");  
 /\*synchronized (daoCincoVar) {  
 daoCincoVar.save(pivote); //Guarda el pivote en la BD  
 }\*/  
   
 return pivote;  
 }  
  
 /\*\*  
 \* De una lista de cincominutalVars  
 \* @param lista una lista de cincominutalVars  
 \* @return variables corregidas  
 \* @throws Exception  
 \*/  
 private List<TxCincominutalVar> corregirAmbientales(List<TxCincominutalVar> lista) throws Exception {  
 List<TxCincominutalVar> varsCorregir = new ArrayList<>();  
 //Obtiene el primer elemento de la lista  
 TxCincominutalVar varPrimeraValida = lista.get(0);  
 //Obtiene el último elemento de la lista  
 TxCincominutalVar varUltimaValida = lista.get(lista.size() - 1);  
 //por cada elemento de la lista excepto el primero y el último  
 for (int i = 1; i < lista.size() - 2; i++) { //AJUSTE PARA QUE FUNCIONE LA INTERPOLACION ERA {lista.size() - 1}   
 //Lo añade a varsCorregir  
 varsCorregir.add(lista.get(i));  
 //varsCorregir es básicamente la lista sin el primer elemento ni el último  
 }  
 //Cada elemento de la lista se cuenta como un un error  
 int errores = varsCorregir.size();  
 //12 = TEMPERATURA DE AGUA DE CIRCULACION AGUA DE MAR  
 //si el tipo de variable es "TEMPERATURA DE AGUA DE CIRCULACION AGUA DE MAR"  
 if (varPrimeraValida.getCatVariable()  
 .getVariableId()  
 .intValue() == 12) {  
 //&& hay entre 1 y 36 errores  
 if (errores > 0 && errores < 37) {  
 //retorna la lista interpolada  
 return interpolacionAmbientales(varsCorregir, varPrimeraValida, varUltimaValida);  
 //(si es TEMPERATURA DE AGUA) && hay más de 36 errores pero menos de 30 días con errores (no toma en cuenta  
 // cambios por horario de verano)  
 } else if (errores > 36 && errores < 30 \* 24 \* 12) { //30 días con errores  
 //debe retornar una lista con valores substituídos  
 return sustitucionAnteriorAmbientales(varsCorregir, 3, 1); //3 = día, 1 = 1 día, sustitucionDiaAnteriorAmbientales  
 //(si es TEMPERATURA DE AGUA) && (la cuenta de errores no es mayor a 0 y menor a 8640)  
 // && hay más de 60 errores (así que en la práctica sólo aplica si hay más de 8640 errores)  
 } else if (errores > 60) {  
 //retorna una lista con valores substituídos por los del mes ANTERIOR  
 List<TxCincominutalVar> response =  
 sustitucionAnteriorAmbientales(varsCorregir, 2,  
 1); //2 = mes, 1 = 1 mes, sustitucionMesAnteriorAmbientales  
 //si el tamaño de la lista es inferior o igual a 0 (por lo tanto, si la lista no contiene elementos  
 // pues no puede ser de tamaño negativo)  
 if (response.size() <= 0) {  
 //subsituye los valores de los cincominutalVar por los equivalentes de la otra planta  
 response = sustitucionPorPlantaAmbientales(varsCorregir);  
 }  
 return response;  
 }  
 //si el tipo de variable NO es "TEMPERATURA DE AGUA DE CIRCULACION AGUA DE MAR"  
 } else {  
 if (errores > 0 && errores < 13) {  
 //Si hay menos de 11 errores (el último valor que es válido se está tomando como error porque es parte de la lista)  
 // devuelve una lista de los valores interpolados  
 return interpolacionAmbientales(varsCorregir, varPrimeraValida, varUltimaValida);  
 //Si los errores contados son más de 12 y menos de 61 estrictamente (5 horas), substituye por el día ANTERIOR  
 //En la norma indica que deberían de ser entre 12 cincominutales y 30 días para variables ambientales y 7 para cromatográficas  
 } else if (errores > 12 && errores < 61) {  
 return sustitucionAnteriorAmbientales(varsCorregir, 3,  
 1); //3 = día, 1 = 1 día, sustitucionDiaAnteriorAmbientales  
 } else if (errores > 60) {  
 //Si hay más de 5 horas (deberían ser 30 días) con valores incorrectos, substituye por los valores del mes pasado  
 List<TxCincominutalVar> response =  
 sustitucionAnteriorAmbientales(varsCorregir, 2,  
 1); //2 = mes, 1 = 1 mes, sustitucionMesAnteriorAmbientales  
 //Si hay más de 5 horas con valores incorrectos && no hay valores del mes ANTERIOR, substituye por los de la otra planta  
 if (response.size() <= 0) {  
 response = sustitucionPorPlantaAmbientales(lista);  
 }  
 return response;  
 }  
 }  
 return varsCorregir;  
 }  
  
 /\*\*  
 \*  
 \* @param lista lista de txCincominutalVar  
 \* @param varPrimeraValida valor del primer elemento de la lista  
 \* @param varB valor del último elemento de la lista  
 \* @return la lista interpolada  
 \* @throws Exception  
 \*/  
 private List<TxCincominutalVar> interpolacionAmbientales(List<TxCincominutalVar> lista,  
 TxCincominutalVar varPrimeraValida,  
 TxCincominutalVar varB) throws Exception {  
 List<TxCincominutalVar> varsCcm = new ArrayList<>();  
 BigDecimal a = varPrimeraValida.getValorFormato();  
 BigDecimal b = varB.getValorFormato();  
 int x = 1;  
 int n = lista.size();  
 //por cada elemento de la lista  
 for (TxCincominutalVar var : lista) {  
 //Las variables a interpolar no deben contener los elementos válidos  
 double valor = JRUtil.interpolacion(x, n, a, b);  
 x++;  
 TxCincominutalVar varCorregida = var; //variable redundante  
 //guarda el resultado de la interpolación  
 varCorregida.setValorNorma(new BigDecimal(valor));  
 varCorregida.setEstatusCn(new BigDecimal(20)); //20 = CORREGIDO POR INTERPOLACION LINEAL  
 varCorregida.setFechaCn(new Date());  
 varCorregida.setFechaUm(new Date());  
 varsCcm.add(varCorregida);  
 }  
 return varsCcm;  
 }  
 /\*  
 private TxCincominutalVar getVarPrimeraValida(CatVariable catVariable) {  
 TxCincominutalVar varPrimeraValida = null;  
 try {  
 //varPrimeraValida = daoCincoVar.getPrimeraValidaCorreccion(valido, noCorregido, catVariable.getVariableId(),PageRequest.of(0, 1)).get(0);  
  
 } catch (Exception ex) {  
 varPrimeraValida = null;  
 }  
 return varPrimeraValida;  
 }\*/  
  
 /\*\*  
 \* @param catVariable objeto (bean) que representa el tipo de variable  
 \* @return devuelve la primera variable del tipo elegido que tenga estatusDn valido y estatusCn validoSinCorregir  
 \*/  
 @SuppressWarnings("Duplicates")  
 private TxCincominutalVar getMinVarPrimeraValida(CatVariable catVariable) { //catVariable podria reemplazarse por variableId y funcionaria igual, es equivalente  
 TxCincominutalVar varPrimeraValida = null;  
 //Guarda el Id del primer cincominutal de la variable que tenga estatusDn = 27 y estatusCn = 28  
 //En su último commit, Ricardo cambió el valor de estatusCn de -1 a 28  
 System.out.printf("Variable: %s\n", catVariable.getNombre());  
 System.out.printf("Planta: %d\n", catPlanta.getPlantaId().intValue());  
 BigDecimal txCincominutalVarId =  
 daoCincoVar.getTxCincominutalVarPrimera(catVariable.getVariableId(), valido,  
 catPlanta.getPlantaId());  
 //Si el cincominutal mencionado existe, lo retorna despues de jalarlo de la base de datos usando su Id  
 if (txCincominutalVarId != null) {  
 varPrimeraValida = daoCincoVar.findById(txCincominutalVarId).orElse(new TxCincominutalVar());  
 } else {  
 System.out.printf("No se encontró primera válida para %s\n", catVariable.getNombre());  
 }  
 return varPrimeraValida;  
 }  
  
 private List<TxCincominutalVar> sustitucionAnteriorAmbientales(List<TxCincominutalVar> lista, Integer unidadTiempo,  
 Integer saltoTemporal) throws Exception {  
 //TODO: Revisar que sí esté haciendo la substitución  
 List<TxCincominutalVar> sustituciones = new ArrayList<>();  
 //por cada elemento de la lista que se recibió  
 for (TxCincominutalVar txVar : lista) {  
  
 CatMesDiaFirma diaAnterior =  
 tool.getMinusCatMesDiaFirma(txVar.getTxCincominutal().getCatMesDiaFirma(), unidadTiempo, saltoTemporal);  
 //nuevoVar nunca se usa  
 TxCincominutalVar txNuevoVar =  
 daoCincoVar.getTxCincoVarAnterior(diaAnterior.getMesDiaFirmaId(),  
 txVar.getCatVariable().getVariableId(), catPlanta.getPlantaId());  
  
 //¿No debería de actualizar y añadir txNuevoVar en lugar de txVar?  
 if (txNuevoVar!=null){  
 txVar.setValorNorma(txNuevoVar.getValorNorma());  
 } else {  
 System.out.printf("No hay valor del día anterior para corrección de variable:\n"  
 + "var: %s\nfechaFirma: %s\n...\n",  
 txVar.getCatVariable().getNombre(), txVar.getTxCincominutal().getCatMesDiaFirma().getFechaFirma().toString());  
 }  
  
 txVar.setEstatusCn(new BigDecimal(21)); //21 = CORREGIDO POR SUSTITUCION DIA ANTERIOR  
 txVar.setFechaCn(new Date());  
 txVar.setFechaUm(new Date());  
 sustituciones.add(txVar);  
 }  
 //TODO: ¿Qué pasa si no hay datos para el día ANTERIOR?  
 //TODO: Si los datos del día ANTERIOR no son válidos, debería jalar los del día ANTERIOR a ese  
 return sustituciones;  
 }  
  
 //TODO: refactorizar para evitar código duplicado con TaskCorreccionCromatograficas  
 @SuppressWarnings("Duplicates")  
 private List<TxCincominutalVar> sustitucionPorPlantaAmbientales(List<TxCincominutalVar> lista) throws Exception {  
 List<TxCincominutalVar> sustituciones = new ArrayList<>();  
 CatPlanta nuevaPlanta = null;  
 if (getCatPlanta().getPlantaId().intValue() == 1) { //No es necesario usar getCatPlanta() porque catPlanta está en la misma clase y es accesible directamente  
 nuevaPlanta = daoCatPlanta.findById(new BigDecimal(2)).orElse(new CatPlanta());  
 }  
 if (getCatPlanta().getPlantaId().intValue() == 2) {  
 nuevaPlanta = daoCatPlanta.findById(new BigDecimal(1)).orElse(new CatPlanta());  
 }  
 for (TxCincominutalVar txVar : lista) {  
  
 //txNuevoVar nunca se usa  
 //Devuelve el cincominutalVar con mismo MesDiaFirma, para la misma variable pero para la otra planta  
 assert nuevaPlanta != null; //para evitar un null pointer exception  
 TxCincominutalVar txNuevoVar =  
 daoCincoVar.getTxCincoVarOtraPlanta(txVar.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId(),  
 txVar.getCatVariable().getVariableId(),  
 nuevaPlanta.getPlantaId());  
 txVar.setValorNorma(txNuevoVar.getValorNorma());  
 txVar.setEstatusCn(new BigDecimal(24)); //24 = CORREGIDO POR SUSTITUCION VALORES OTRA PLANTA  
 txVar.setFechaCn(new Date());  
 txVar.setFechaUm(new Date());  
 sustituciones.add(txVar);  
 }  
 return sustituciones;  
 }  
  
  
 public CatPlanta getCatPlanta() {  
 return catPlanta;  
 }  
  
 public void setCatPlanta(CatPlanta catPlanta) {  
 this.catPlanta = catPlanta;  
 }  
  
 public CatVariable getCatVariable() {  
 return catVariable;  
 }  
  
 public void setCatVariable(CatVariable catVariable) {  
 this.catVariable = catVariable;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\dyc\TaskCorreccionCromatrograficas.java

package mx.mbi.dgc.business.dyc;  
  
import java.math.BigDecimal;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import mx.mbi.dgc.dao.DAOCatPlanta;  
import mx.mbi.dgc.dao.DAOTxCincominutal;  
import mx.mbi.dgc.dao.negocio.DAOCincoVar;  
import mx.mbi.dgc.dao.negocio.DAODyCNorma;  
import mx.mbi.dgc.dao.negocio.DAOVariable;  
import mx.mbi.dgc.dao.negocio.DaoNegocioTxCincominutalVar;  
import mx.mbi.dgc.dominio.DominioDyCNorma;  
import mx.mbi.dgc.jpa.entity.CatMesDiaFirma;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.jpa.entity.CatVariable;  
import mx.mbi.dgc.jpa.entity.TxCincominutal;  
import mx.mbi.dgc.jpa.entity.TxCincominutalVar;  
import mx.mbi.dgc.utils.DominioTool;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.annotation.Scope;  
import org.springframework.stereotype.Component;  
  
@Component  
@Scope("prototype")  
public class TaskCorreccionCromatrograficas implements Runnable {  
 private class EstatusNormaVariable{  
 BigDecimal estatusDn = null;  
 BigDecimal estatusCn = null;  
 }  
 private enum Direcciones {  
 ANTERIOR,  
 SIGUIENTE;  
 }  
  
 private static final Logger LOG = LogManager.getLogger(BusinessCorreccionNorma.class);  
  
 @Autowired  
 private DAOVariable daoCatVariable;  
  
 @Autowired  
 private DAOCincoVar daoCincoVar;  
  
 @Autowired  
 private DAOTxCincominutal daoTxCincominutal;  
  
 @Autowired  
 private DAOCatPlanta daoCatPlanta;  
  
 @Autowired  
 private DominioTool tool;  
  
 @Autowired  
 private DAODyCNorma daoDyCNorma;  
  
 @Autowired  
 private DaoNegocioTxCincominutalVar daoNegocioTxCincominutalVar;  
  
 private CatPlanta catPlanta;  
 private CatVariable catVariable;  
 private DominioDyCNorma dom;  
  
 private static BigDecimal noCorregido = new BigDecimal("-1"); //-1 = NO CORREGIDO  
 private static BigDecimal valido = new BigDecimal("27"); //27 = VALIDO  
 private static BigDecimal congeladoValido = new BigDecimal("32"); //27 = congelado VALIDO  
 private static BigDecimal pivote = new BigDecimal("1"); //1 = VALIDO  
 private static BigDecimal validoSinCorregir = new BigDecimal("28"); //28 = VALIDO SIN CORREGIR  
  
  
 public TaskCorreccionCromatrograficas() {  
 super();  
 }  
  
 @Override  
 public void run() {  
 this.catPlanta = daoCatPlanta.findById(catVariable.getCatPlanta().getPlantaId()).orElse(new CatPlanta());  
  
 //primer cincominutal de estatusDn = 27 y estatusCn = 28  
 TxCincominutalVar varPrimeraValida = getMinVarPrimeraValida(catVariable);  
  
 try {  
 if (varPrimeraValida != null) {  
 System.out.println("Primera valida encontrada, buscando pivotes");  
  
 // Recupera la lista de cincominutalVars de {estatusCn = -1} y de cincominutal estrictamente posterior  
 // al primer cincominutalVar de {estatusDn = 27} y {estatusCn = 28}, de la misma planta  
 // y la misma variable  
 List<TxCincominutalVar> listaTxCincoVarNoCorregidos =  
 daoCincoVar.getListaAmbientales(noCorregido,  
 getCatVariable().getVariableId(),  
 varPrimeraValida.getTxCincominutal().getCincominutalId(),  
 catPlanta.getPlantaId());;  
  
  
 System.out.printf("Se encontraron %d registros sin corregir para la variable %s\n",  
 listaTxCincoVarNoCorregidos.size(), catVariable.getNombre());  
  
 //corregirALaRicardo(varPrimeraValida, listaTxCincoVarNoCorregidos);  
  
 TxCincominutalVar buffer = null;  
 int corregidos = 0;  
 ArrayList<TxCincominutalVar> tuplaACorregir = new ArrayList<>();  
  
 //Los cincominutales se corrigen según el número de valores erróneos consecutivos  
 for (TxCincominutalVar cincominutal : listaTxCincoVarNoCorregidos) {  
 if (buffer == null) {  
 System.out.println("Guardando primer cincominutal a corregir");  
 //Esto ocurre al principio de cada tupla  
 tuplaACorregir.add(cincominutal);  
 } else if (sonConsecutivos(buffer, cincominutal)) {  
 System.out.printf("Agregando siguiente cincominutal a corregir, #%d\n", ++corregidos);  
 tuplaACorregir.add(cincominutal);  
 } else {  
 System.out.printf("Corrigiendo %d cincominutales consecutivos\n", tuplaACorregir.size());  
 corregir(tuplaACorregir);  
 //Crea una nueva tupla a corregir, limpia el buffer.  
 tuplaACorregir = new ArrayList<>();  
 buffer = null;  
 continue;  
 }  
 buffer = cincominutal;  
 }  
 if (tuplaACorregir.size() > 0) {  
 corregir(tuplaACorregir);  
 }  
 } else {  
 System.out.println("No Existen datos para realizar la correccion en TaskCorreccionAmbientales");  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 /\*\*  
 \* @param catVariable objeto (bean) que representa el tipo de variable  
 \* @return devuelve la primera variable del tipo elegido que tenga estatusDn valido y estatusCn validoSinCorregir  
 \*/  
 @SuppressWarnings("Duplicates")  
 private TxCincominutalVar getMinVarPrimeraValida(CatVariable catVariable) { //catVariable podria reemplazarse por variableId y funcionaria igual, es equivalente  
 TxCincominutalVar varPrimeraValida = null;  
 //Guarda el Id del primer cincominutal de la variable que tenga estatusDn = 27 y estatusCn = 28  
 //En su último commit, Ricardo cambió el valor de estatusCn de -1 a 28  
 System.out.printf("Variable: %s\n", catVariable.getNombre());  
 System.out.printf("Planta: %d\n", catPlanta.getPlantaId().intValue());  
 BigDecimal txCincominutalVarId =  
 daoCincoVar.getTxCincominutalVarPrimera(catVariable.getVariableId(), valido,  
 catPlanta.getPlantaId());  
 //Si el cincominutal mencionado existe, lo retorna despues de jalarlo de la base de datos usando su Id  
 if (txCincominutalVarId != null) {  
 varPrimeraValida = daoCincoVar.findById(txCincominutalVarId).orElse(new TxCincominutalVar());  
 }  
 return varPrimeraValida;  
 }  
  
 private Boolean sonConsecutivos(TxCincominutalVar primero, TxCincominutalVar segundo) {  
 BigDecimal mesDiaFirmaPrimero = primero.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId();  
 BigDecimal mesDiaFirmaSegundo = segundo.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId();  
  
 return mesDiaFirmaPrimero.equals(mesDiaFirmaSegundo.subtract(BigDecimal.ONE));  
 }  
  
 private void corregir(ArrayList<TxCincominutalVar> tuplaACorregir) {  
 TxCincominutalVar primerValorInvalido = tuplaACorregir.get(0);  
 TxCincominutalVar ultimoValorInvalido = tuplaACorregir.get(tuplaACorregir.size() - 1);  
 TxCincominutalVar primerValorValido = getValorConsecutivo(primerValorInvalido, Direcciones.ANTERIOR);  
 TxCincominutalVar ultimoValorValido = getValorConsecutivo(ultimoValorInvalido, Direcciones.SIGUIENTE);  
  
 if (primerValorValido == null) {  
 primerValorValido = primerValorInvalido;  
 }  
 if (ultimoValorValido == null) {  
 ultimoValorValido = ultimoValorInvalido;  
 }  
 if (ultimoValorValido.getEstatusDn().equals(BigDecimal.ONE)) {  
 actualizarPivote(ultimoValorValido);  
 }  
  
 if (!primerValorValido.getEstatusDn().equals(valido)&&!ultimoValorValido.getEstatusDn().equals(valido)&&  
 !ultimoValorValido.getEstatusDn().equals(BigDecimal.ONE)) {  
 System.out.printf("Estatus primer valor: %d\nEstatus ultimo valor: %d\n",  
 primerValorValido.getEstatusDn().intValue(),  
 ultimoValorValido.getEstatusDn().intValue());  
 System.out.printf("No se puede corregir tupla de %s para el cincominutal por falta de datos validos %s\n",  
 catVariable.getNombre(), tuplaACorregir.get(0).getTxCincominutal().getCatMesDiaFirma().getFechaFirma().toString());  
 if (!primerValorValido.getEstatusDn().equals(valido)) {  
 tuplaACorregir.add(0, primerValorValido);  
 }  
 if (!ultimoValorValido.getEstatusDn().equals(valido)) {  
 tuplaACorregir.add(ultimoValorValido);  
 }  
 corregir(tuplaACorregir);  
 return;  
 }  
  
 if (!primerValorValido.equals(tuplaACorregir.get(0))) {  
 tuplaACorregir.add(0, primerValorValido);  
 }  
  
 if (!ultimoValorValido.equals(tuplaACorregir.get(tuplaACorregir.size()-1))) {  
 tuplaACorregir.add(ultimoValorValido);  
 }  
  
 try {  
 corregirALaAntigua(primerValorValido,tuplaACorregir);  
 //System.out.printf("Guardando %d valores corregidos", corregidos.size());  
 //daoCincoVar.saveAll(corregidos);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 }  
  
 private TxCincominutalVar getValorConsecutivo(TxCincominutalVar valor, Direcciones direccion) {  
 BigDecimal mesDiaFirmaId = valor.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId();  
 BigDecimal variableId = valor.getCatVariable().getVariableId();  
 BigDecimal catPlanta = this.catPlanta.getPlantaId();  
 TxCincominutalVar valorConsecutivo;  
  
 try {  
 if (direccion == Direcciones.ANTERIOR) {  
 valorConsecutivo = daoCincoVar.getTxCincoVarAnterior(mesDiaFirmaId.subtract(BigDecimal.ONE), variableId, catPlanta);  
 } else {  
 valorConsecutivo = daoCincoVar.getTxCincoVarAnterior(mesDiaFirmaId.add(BigDecimal.ONE), variableId, catPlanta);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 valorConsecutivo = null;  
 }  
 return valorConsecutivo;  
 }  
  
  
 @SuppressWarnings("Duplicates")  
 private void corregirALaAntigua(TxCincominutalVar varPrimeraValida, List<TxCincominutalVar> listaTxCincoVarNoCorregidos)  
 throws Exception{  
 System.out.printf("Tamano de la lista: %d\n", listaTxCincoVarNoCorregidos.size());  
 List<TxCincominutalVar> tuplaCorreccion = new ArrayList<>();  
 int variablesProcesadas = 0;  
 //si la lista existe y no esta vacia  
 if (listaTxCincoVarNoCorregidos != null && listaTxCincoVarNoCorregidos.size() > 0) {  
 //Se añade el primer cincominutal de estatusDn = 27 y estatusCn = 28 a tuplaCorreccion.  
 tuplaCorreccion.add(varPrimeraValida);  
 //Tiempo de inicio de proceso de corrección  
 Date contadorDeTiempoInicio = new Date();  
 //por cada elemento de la lista de cincominutalVars de {estatusCn = -1} posteriores a la primera válida  
 for (TxCincominutalVar var : listaTxCincoVarNoCorregidos) {  
 System.out.printf("Valor del cincominutal %d: %f, estatus: %d\n",  
 var.getTxCincominutal().getCincominutalId().intValue(),  
 var.getValorFormato().floatValue(),  
 var.getEstatusDn().intValue());  
 //Para mostrar la velocidad de correccion de variables  
 if (variablesProcesadas>=1000){  
 Date contadorDeTiempoFinal = new Date();  
 Long contadorDeTiempo = contadorDeTiempoFinal.getTime()-contadorDeTiempoInicio.getTime();  
 System.out.printf("Tardo %d milisegundos en procesar " +  
 "los ultimos %d registros\n", contadorDeTiempo, variablesProcesadas);  
 float velocidad = variablesProcesadas/(contadorDeTiempo/1000f);  
 System.out.printf("Se procesan a %f variables por segundo\n", velocidad);  
 //vuelve a empezar la cuenta para medir la velocidad de las siguientes 1000 variables  
 variablesProcesadas = 0;  
 contadorDeTiempoInicio = contadorDeTiempoFinal;  
 }  
  
 //Se añade el elemento a tuplaCorreccion.  
 tuplaCorreccion.add(var);  
  
 //Si el estatus de Deteccion de Norma de algún elemento de la lista de no corregidos  
 // posteriores al primer cincominutalVar valido es "valido"  
 if (var.getEstatusDn().intValue() == valido.intValue()) {//es valida  
 //Guarda el primer elemento de tuplaCorreccion (varPrimeraValida) en la BD  
 // tal que {valorNorma = valorFormato}, {estatusCn = validoSinCorregir (28)}  
 // {fechaUm = fecha actual} y {fechaCn = fechaUm}  
 actualizarPrimeraValida(tuplaCorreccion.get(0));  
  
 //(si es válido &&) si tuplaCorreccion contiene 2 elementos, lo cuál significa que ambos son válidos  
 if (tuplaCorreccion.size() == 2) {  
 //TODO: ¿cambiar el estatus a 28?  
 for (TxCincominutalVar elementoValido :  
 tuplaCorreccion) {  
 EstatusNormaVariable estatus = new EstatusNormaVariable();  
 estatus.estatusCn = validoSinCorregir;  
 actualizarVariable(elementoValido, estatus);  
 }  
 //Borra tuplaCorreccion  
 tuplaCorreccion = new ArrayList<>();  
 //Añade el elemento actual (que es válido) de la lista de no corregidos,  
 // al principio de tuplaCorreccion  
 tuplaCorreccion.add(var);  
 //Termina la iteración actual y pasa al elemento no corregido SIGUIENTE  
 variablesProcesadas++;  
 continue;  
 }  
  
 //Si hay un cincominutalVar con estatusDn "valido", aplica la corrección a tuplaCorreccion  
 // borra tuplaCorreccion  
 // y añade el cincominutalVar (que es válido)a la nueva tuplaCorreccion  
  
  
 List<TxCincominutalVar> corregidos = corregirCromas(tuplaCorreccion);  
  
 /\*  
 System.out.printf("Guardando %d valores\n", corregidos.size());  
 for (TxCincominutalVar cincominutalVar :  
 corregidos) {  
 System.out.printf("Guardando %f...\n", cincominutalVar.getValorNorma().floatValue());  
 }  
 daoCincoVar.saveAll(corregidos);\*/  
 /\*  
 synchronized (daoCincoVar) {  
 daoCincoVar.saveAll(corregidos);  
 }\*/  
 tuplaCorreccion = new ArrayList<>();  
 tuplaCorreccion.add(var);  
 tuplaCorreccion.add(var);  
 }  
 variablesProcesadas++;  
 }  
  
 //Después de revisar todos los cincominutalVars no corregidos, guarda el primero en la base de datos  
 // con {estatusCn = validoSinCorregir} y fechaUm y fechaCm actuales  
 // tuplaCorreccion siempre tiene al menos un elemento así que siempre corre.  
 //en su commit del 14 de enero RTC cambió de >1 a >0  
 if(tuplaCorreccion.size() > 0){  
 actualizarPrimeraValida(tuplaCorreccion.get(0));  
 }  
 }  
 }  
  
  
 /\*\*  
 \* Es lo mismo que actualizarPivote solo que pone el estatusCn en validoSinCorregir en lugar de valido.  
 \* Se pueden refactorizar a un solo método.  
 \* @param primeraValida un txCincominutalVar que se supone es el primer cincominutalVar de estatusDn = 27 y estatusCn = 28  
 \*/  
 private void actualizarPrimeraValida(TxCincominutalVar primeraValida){  
 //TODO: Refactorizar a actualizarCincominutalVar()  
 primeraValida.setValorNorma(primeraValida.getValorFormato());  
 primeraValida.setEstatusCn(validoSinCorregir);  
 primeraValida.setFechaUm(new Date());  
 primeraValida.setFechaCn(primeraValida.getFechaUm());  
 daoCincoVar.save(primeraValida);  
  
 /\*synchronized (daoCincoVar) {  
 daoCincoVar.save(primeraValida);  
 }\*/  
 }  
  
 /\*\*  
 \* Version refactorizada (mas general) de actualizarPrimeraValida y ActualizarPivote  
 \* @param variable una cincominutalVar  
 \* @param estatus objeto que contiene los valores a cambiar, los valores que no se cambian deben dejarse como null.  
 \*/  
 private void actualizarVariable(TxCincominutalVar variable, EstatusNormaVariable estatus){  
 variable.setValorNorma(variable.getValorFormato());  
 variable.setFechaUm(new Date());  
 if (estatus.estatusDn!=null){  
 variable.setEstatusDn(estatus.estatusDn);  
 variable.setFechaDn(variable.getFechaUm());  
 }  
 if (estatus.estatusCn!=null){  
 variable.setEstatusCn(estatus.estatusCn);  
 variable.setFechaCn(variable.getFechaUm());  
 }  
 daoCincoVar.save(variable);  
 }  
  
 /\* ----------- ANTIGUA VERSIÓN de run():  
 List<TxCincominutal> tuplaCorreccion = null;  
 try {  
 TxCincominutal varCincoPrimerValido = null;  
 try {  
 List<TxCincominutalVar> listPivotes = daoCincoVar.obtenerPivotes(catPlanta.getPlantaId());  
 for (TxCincominutalVar txCincominutalVar : listPivotes) {  
 actualizarPivote(txCincominutalVar);  
 }  
 varCincoPrimerValido = getPrimerTxCincominutalVarValido(); //C = Cromatograficas  
 tuplaCorreccion = new ArrayList<TxCincominutal>();  
 tuplaCorreccion.add(varCincoPrimerValido);  
 List<TxCincominutal> listaTxCincominutal =  
 daoDyCNorma.getCincominutalesCorregirCroma(varCincoPrimerValido.getCincominutalId(),  
 catPlanta.getPlantaId(), noCorregido, "C");  
 if (listaTxCincominutal != null && listaTxCincominutal.size() > 0) {  
 int control = 0;  
 for (TxCincominutal ccm : listaTxCincominutal) {  
 List<TxCincominutalVar> varsCromas = findCromas(ccm);  
 tuplaCorreccion.add(ccm);  
 boolean status = false;  
 for (TxCincominutalVar valVar : varsCromas) {  
 if ((valVar.getEstatusDn().intValue() == valido.intValue()) || (valVar.getEstatusDn().intValue() == congeladoValido.intValue())) {  
 status = true;  
 break;  
 }  
 }  
 if (status) {  
 for (TxCincominutalVar varCroma : varCincoPrimerValido.getTxCincominutalVars()) {  
 varCroma.setValorNorma(varCroma.getValorFormato());  
 varCroma.setEstatusCn(validoSinCorregir);  
 varCroma.setFechaUm(new Date());  
 varCroma.setFechaCn(varCroma.getFechaUm());  
 synchronized (daoCincoVar) {  
 daoCincoVar.save(varCroma);  
 }  
 }  
 if (tuplaCorreccion.size() == 2) {  
 varCincoPrimerValido = tuplaCorreccion.get(1);  
 tuplaCorreccion = new ArrayList<TxCincominutal>();  
 tuplaCorreccion.add(varCincoPrimerValido);  
 } else {  
 int errores = maxErrors(tuplaCorreccion);  
 List<TxCincominutalVar> lista = null;  
 List<TxCincominutalVar> listaUpdates = new ArrayList<>();  
 Map<CatVariable, List<TxCincominutalVar>> listaTxCincominutalVar = obtenerTxCincominutalesVar(tuplaCorreccion);  
 for (Map.Entry<CatVariable, List<TxCincominutalVar>> entry : listaTxCincominutalVar.entrySet()) {  
 List<TxCincominutalVar> corregidos = corregirCromas((List<TxCincominutalVar>)entry.getValue(), errores);  
 for(TxCincominutalVar txUpdate : corregidos){  
 TxCincominutalVar varStatusUpdate = daoCincoVar.findById(txUpdate.getCincominutalVarId()).orElse(new TxCincominutalVar());  
 varStatusUpdate.setValorNorma(txUpdate.getValorNorma());  
 varStatusUpdate.setEstatusCn(txUpdate.getEstatusCn());  
 varStatusUpdate.setFechaUm(new Date());  
 varStatusUpdate.setFechaCn(varStatusUpdate.getFechaUm());  
 listaUpdates.add(varStatusUpdate);  
 }  
 }  
  
 synchronized (daoCincoVar) {  
 daoCincoVar.saveAll(listaUpdates);  
 }  
 varCincoPrimerValido =  
 tuplaCorreccion.get(tuplaCorreccion.size() - 1); //último cinco es el nuevo válido  
 tuplaCorreccion = new ArrayList<TxCincominutal>();  
 tuplaCorreccion.add(varCincoPrimerValido);  
 }  
 }  
 control++;  
 if (control % 1000 == 0) {  
 System.gc();  
 }  
 }  
 } else {  
 System.out.println("No existen varaibles cromatograficas a corregir");  
 }  
  
 } catch (Exception ex) {  
 System.out.println("No hay informacion a corregir: ");  
 ex.printStackTrace();  
 }  
  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 \*/  
 private Map<CatVariable, List<TxCincominutalVar>> obtenerTxCincominutalesVar(List<TxCincominutal> tuplaCorreccion) {  
 Map<CatVariable, List<TxCincominutalVar>> retorno = new HashMap<>();  
 List<TxCincominutalVar> listaActual = null;  
 for (TxCincominutal cincoSave : tuplaCorreccion) {  
 for (TxCincominutalVar txcvar : cincoSave.getTxCincominutalVars()) {  
 if (txcvar.getCatVariable()  
 .getTipo()  
 .equals("C")) {  
 if (retorno.get(txcvar.getCatVariable()) == null) {  
 listaActual = new ArrayList<>();  
 retorno.put(txcvar.getCatVariable(), listaActual);  
 }  
 listaActual = retorno.get(txcvar.getCatVariable());  
 listaActual.add(txcvar);  
 Collections.sort(listaActual);  
 retorno.put(txcvar.getCatVariable(), listaActual);  
 }  
 }  
 }  
 return retorno;  
 }  
  
 private TxCincominutalVar actualizarPivote(TxCincominutalVar pivote) {  
 pivote.setValorNorma(pivote.getValorFormato());  
 pivote.setEstatusDn(valido);  
 pivote.setFechaUm(new Date());  
 pivote.setFechaDn(pivote.getFechaUm());  
 daoCincoVar.save(pivote);  
  
 return pivote;  
 }  
  
 private List<TxCincominutalVar> interpolacionAmbientales(List<TxCincominutalVar> lista,  
 TxCincominutalVar varPrimeraValida,  
 TxCincominutalVar varB) throws Exception {  
 List<TxCincominutalVar> varsCcm = new ArrayList<>();  
 BigDecimal a = varPrimeraValida.getValorFormato();  
 BigDecimal b = varB.getValorFormato();  
 int x = 1;  
 int n = lista.size();  
 //System.out.println("########## a: "+varPrimeraValida.getCincominutalVarId()+" Value: "+varPrimeraValida.getValor());  
 for (TxCincominutalVar var : lista) {  
 double valor = JRUtil.interpolacion(x, n, a, b);  
 x++;  
 TxCincominutalVar varCorregida = var;  
 varCorregida.setCincominutalVarId(var.getCincominutalVarId());  
 varCorregida.setValorNorma(new BigDecimal(valor));  
 varCorregida.setEstatusCn(new BigDecimal(20)); //20 = CORREGIDO POR INTERPOLACION LINEAL  
 //System.out.println("##### Nuevo Valor var: "+var.getCincominutalVarId()+" Value anterior: "+var.getValor()+" Nuevo: "+varCorregida.getValor());  
 varsCcm.add(varCorregida);  
 }  
 //System.out.println("########## b: "+varB.getCincominutalVarId()+" Value: "+varB.getValor());  
 System.out.printf("Se interpolaron %d valores\n", varsCcm.size());  
 return varsCcm;  
 }  
  
 private List<TxCincominutalVar> sustitucionAnteriorAmbientales(List<TxCincominutalVar> lista, Integer unidadTiempo,  
 Integer saltoTemporal) throws Exception {  
 List<TxCincominutalVar> sustituciones = new ArrayList<>();  
 for (TxCincominutalVar txVar : lista) {  
 CatMesDiaFirma diaAnterior =  
 tool.getMinusCatMesDiaFirma(txVar.getTxCincominutal().getCatMesDiaFirma(), unidadTiempo, saltoTemporal);  
 TxCincominutalVar txNuevoVar =  
 daoCincoVar.getTxCincoVarAnterior(diaAnterior.getMesDiaFirmaId(),  
 txVar.getCatVariable().getVariableId(), catPlanta.getPlantaId());  
 sustituciones.add(txNuevoVar);  
 }  
 return sustituciones;  
 }  
  
 private List<TxCincominutalVar> sustitucionPorPlantaAmbientales(List<TxCincominutalVar> lista) throws Exception {  
 List<TxCincominutalVar> sustituciones = new ArrayList<>();  
 CatPlanta nuevaPlanta = null;  
 if (getCatPlanta().getPlantaId().intValue() == 1) {  
 nuevaPlanta = daoCatPlanta.findById(new BigDecimal(2)).orElse(new CatPlanta());  
 }  
 if (getCatPlanta().getPlantaId().intValue() == 2) {  
 nuevaPlanta = daoCatPlanta.findById(new BigDecimal(1)).orElse(new CatPlanta());  
 }  
 for (TxCincominutalVar txVar : lista) {  
 TxCincominutalVar txNuevoVar =  
 daoCincoVar.getTxCincoVarOtraPlanta(txVar.getTxCincominutal()  
 .getCatMesDiaFirma()  
 .getMesDiaFirmaId(),  
 txVar.getCatVariable().getVariableId(), nuevaPlanta.getPlantaId());  
 sustituciones.add(txNuevoVar);  
 }  
 return sustituciones;  
 }  
  
 public CatPlanta getCatPlanta() {  
 return catPlanta;  
 }  
  
 public void setCatPlanta(CatPlanta catPlanta) {  
 this.catPlanta = catPlanta;  
 }  
  
 public CatVariable getCatVariable() {  
 return catVariable;  
 }  
  
 public void setCatVariable(CatVariable catVariable) {  
 this.catVariable = catVariable;  
 }  
  
 private List<TxCincominutalVar> findCromas(TxCincominutal ccm) {  
 List<TxCincominutalVar> lista = new ArrayList<>();  
 for (TxCincominutalVar ccmVar : ccm.getTxCincominutalVars()) {  
 if (ccmVar.getCatVariable()  
 .getTipo()  
 .equalsIgnoreCase("C")) { //C = Cromatograficas  
 lista.add(ccmVar);  
 }  
 }  
 return lista;  
 }  
  
 private int maxErrors(List<TxCincominutal> lista) {  
 int maxErrores = 0;  
 for (TxCincominutalVar txVar : lista.get(0).getTxCincominutalVars()) {  
 if (txVar.getCatVariable().getTipo().equals("C")) {  
 int error = 0;  
 for (TxCincominutal cinco : lista) {  
 for (TxCincominutalVar auxVar : cinco.getTxCincominutalVars()) {  
 if (txVar.getCatVariable()  
 .getVariableId()  
 .intValue() == auxVar.getCatVariable()  
 .getVariableId()  
 .intValue()) {  
 if (auxVar.getEstatusDn().intValue() != valido.intValue()) {  
 error++;  
 }  
 }  
 }  
 }  
 if (error > maxErrores) {  
 maxErrores = error;  
 }  
 }  
 }  
 return maxErrores;  
 }  
  
 @SuppressWarnings("Duplicates")  
 private List<TxCincominutalVar> corregirCromas(List<TxCincominutalVar> lista) throws Exception {  
 ArrayList<TxCincominutalVar> varsCorregir = new ArrayList<>();  
 //Obtiene el primer elemento de la lista  
 TxCincominutalVar varPrimeraValida = lista.get(0);  
 //Obtiene el último elemento de la lista  
 TxCincominutalVar varUltimaValida = lista.get(lista.size() - 1);  
 //por cada elemento de la lista excepto el primero y el último  
 for (int i = 1; i < lista.size() - 2; i++) { //AJUSTE PARA QUE FUNCIONE LA INTERPOLACION ERA {lista.size() - 1}  
 //Lo añade a varsCorregir  
 varsCorregir.add(lista.get(i));  
 //varsCorregir es básicamente la lista sin el primer elemento ni el último  
 }  
 //Cada elemento de la lista se cuenta como un un error  
 int errores = varsCorregir.size();  
 ArrayList<CatVariable> variables = new ArrayList<>(daoCatVariable.findAllByTipoAndCatPlanta("C", catPlanta));  
 System.out.printf("Variables Cromatograficas para planta #%d: %d\n",  
 catPlanta.getPlantaId().intValue(),  
 variables.size());  
 ArrayList<TxCincominutalVar> valoresCorregidos = new ArrayList<>();  
 ArrayList<TxCincominutalVar> listaTxCincoVarACorregir = new ArrayList<>();  
  
 if (errores > 0 && errores < 13) {  
 //Si hay menos de 11 errores (el último valor que es válido se está tomando como error porque es parte de la lista)  
 // devuelve una lista de los valores interpolados  
  
 for (CatVariable variableCroma : variables) {  
 /\*  
 System.out.printf("Corrigiendo variables cromatograficas por interpolacion: %s\n", variableCroma.getNombre());  
 List<TxCincominutalVar> listaTxCincoVarACorregir =  
 daoCincoVar.obtenerTxCincominutalVarEntreCincominutales(  
 catPlanta.getPlantaId(),  
 variableCroma.getVariableId(),  
 varPrimeraValida.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId(),  
 varUltimaValida.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId()  
 );  
 TxCincominutalVar primeraValida = listaTxCincoVarACorregir.get(0);  
 listaTxCincoVarACorregir.remove(0);  
 TxCincominutalVar ultimaValida = listaTxCincoVarACorregir.get(listaTxCincoVarACorregir.size() - 1);  
 listaTxCincoVarACorregir.remove(listaTxCincoVarACorregir.size() - 1);  
 valoresCorregidos = interpolacionAmbientales(listaTxCincoVarACorregir, primeraValida, ultimaValida);  
 System.out.printf("Guardando %d valores...\n", valoresCorregidos.size());  
  
 int indice = 0;  
  
 for (TxCincominutalVar corregido : valoresCorregidos) {  
 corregido.setCatVariable(variableCroma);  
 }\*/  
 for (TxCincominutalVar cincominutal : lista) {  
 BigDecimal mesDiaFirmaId = cincominutal.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId();  
 BigDecimal variableId = variableCroma.getVariableId();  
 TxCincominutalVar variable = daoCincoVar.getTxCincoVarAnterior(mesDiaFirmaId, variableId, catPlanta.getPlantaId());  
 variable.setCatVariable(variableCroma);  
 listaTxCincoVarACorregir.add(variable);  
 }  
  
 TxCincominutalVar primeraValida = listaTxCincoVarACorregir.get(0);  
 listaTxCincoVarACorregir.remove(0);  
 TxCincominutalVar ultimaValida = listaTxCincoVarACorregir.get(listaTxCincoVarACorregir.size() - 1);  
 listaTxCincoVarACorregir.remove(listaTxCincoVarACorregir.size() - 1);  
 valoresCorregidos = new ArrayList<>(interpolacionAmbientales(listaTxCincoVarACorregir, primeraValida, ultimaValida));  
 System.out.printf("Guardando %d valores...\n", valoresCorregidos.size());  
 daoCincoVar.saveAll(valoresCorregidos);  
 }  
 //Si los errores contados son más de 12 y menos de 61 estrictamente (5 horas), substituye por el día ANTERIOR  
 //En la norma indica que deberían de ser entre 12 cincominutales y 30 días para variables ambientales y 7 para cromatográficas  
 } else if (errores > 12 && errores < 61) {  
 System.out.printf("Corrigiendo variables cromatograficas por sustiucion: %s", catVariable.getNombre());  
 for (CatVariable variableCroma : variables) {  
 listaTxCincoVarACorregir = new ArrayList<>(  
 daoCincoVar.obtenerTxCincominutalVarEntreCincominutales(  
 catPlanta.getPlantaId(),  
 variableCroma.getVariableId(),  
 varPrimeraValida.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId(),  
 varUltimaValida.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId()  
 ));  
 listaTxCincoVarACorregir.remove(0);  
 listaTxCincoVarACorregir.remove(listaTxCincoVarACorregir.size() - 1);  
 valoresCorregidos = new ArrayList<>(sustitucionAnteriorAmbientales(listaTxCincoVarACorregir, 3,  
 1)); //3 = día, 1 = 1 día, sustitucionDiaAnteriorAmbientales  
 System.out.printf("Guardando %d valores...\n", valoresCorregidos.size());  
  
 daoCincoVar.saveAll(valoresCorregidos);  
 }  
 } else if (errores > 60) {  
 //Si hay más de 5 horas (deberían ser 30 días) con valores incorrectos, substituye por los valores del mes pasado  
 for (CatVariable variableCroma : variables) {  
 listaTxCincoVarACorregir = new ArrayList<>(  
 daoCincoVar.obtenerTxCincominutalVarEntreCincominutales(  
 catPlanta.getPlantaId(),  
 variableCroma.getVariableId(),  
 varPrimeraValida.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId(),  
 varUltimaValida.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId()  
 ));  
 TxCincominutalVar primeraValida = listaTxCincoVarACorregir.get(0);  
 listaTxCincoVarACorregir.remove(0);  
 TxCincominutalVar ultimaValida = listaTxCincoVarACorregir.get(listaTxCincoVarACorregir.size() - 1);  
 listaTxCincoVarACorregir.remove(listaTxCincoVarACorregir.size() - 1);  
 valoresCorregidos = new ArrayList<>(sustitucionAnteriorAmbientales(varsCorregir, 2,  
 1)); //2 = mes, 1 = 1 mes, sustitucionMesAnteriorAmbientales  
  
 //Si hay más de 5 horas con valores incorrectos && no hay valores del mes ANTERIOR, substituye por los de la otra planta  
 if (valoresCorregidos.size() <= 0) {  
 valoresCorregidos = new ArrayList<>(sustitucionPorPlantaAmbientales(lista));  
 }  
  
 System.out.printf("Guardando %d valores...\n", valoresCorregidos.size());  
  
 daoCincoVar.saveAll(valoresCorregidos);  
 }  
 int indice = 0;  
 if (valoresCorregidos != null && valoresCorregidos.size() > 0) {  
  
 for (TxCincominutalVar valor : valoresCorregidos) {  
 if (valor.equals(null)) {  
 valoresCorregidos.remove(indice);  
 }  
 indice++;  
 }  
 }  
 }  
  
 daoCincoVar.flush();  
  
 return varsCorregir;  
  
 /\*  
 System.out.println("corregirCromas.lista : " + lista.size() + " :errores : " + errores);  
 List<TxCincominutalVar> varsCorregir = new ArrayList<>();  
 TxCincominutalVar varPrimeraValida = lista.get(0);  
 TxCincominutalVar varB = lista.get(lista.size() - 1);  
 for (int i = 1; i < lista.size() - 1; i++) {  
 TxCincominutalVar txVar = lista.get(i);  
 varsCorregir.add(txVar);  
 }  
 if (errores > 0 && errores < 13) {  
 return interpolacionAmbientales(varsCorregir, varPrimeraValida, varB);  
 } else if (errores > 12 && errores < 61) {  
 return sustitucionAnteriorAmbientales(varsCorregir, 3,  
 1); //3 = día, 1 = 1 día, sustitucionDiaAnteriorAmbientales  
 } else if (errores > 60) {  
 /\*List<TxCincominutalVar> response =  
 sustitucionAnteriorAmbientales(varsCorregir, 2,  
 1); //2 = mes, 1 = 1 mes, sustitucionMesAnteriorAmbientales  
 if (response.size() <= 0) {  
 response = sustitucionPorPlantaAmbientales(lista);  
 response = lista;  
 }  
 return response;  
 return varsCorregir;  
 }  
 return varsCorregir;  
 \*/  
 }  
  
  
 public DominioDyCNorma getDom() {  
 return dom;  
 }  
  
 public void setDom(DominioDyCNorma dom) {  
 this.dom = dom;  
 }  
  
 private TxCincominutal getPrimerTxCincominutalVarValido() {  
 TxCincominutal retorno = null;  
 BigDecimal idCincominutal = daoCincoVar.getCincoPrimerValido(valido, "C", catPlanta.getPlantaId(), noCorregido);  
 System.out.println("TaskCorrecionCromatrograficas getPrimerTxCincominutalVarValido: " + idCincominutal);  
 retorno = daoTxCincominutal.obtenTxCincominutal(idCincominutal);  
 System.out.println("TaskCorrecionCromatrograficas getPrimerTxCincominutalVarValido: " + retorno);  
 return retorno;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\dyc\TaskSaveDeteccion.java

package mx.mbi.dgc.business.dyc;  
  
import java.math.BigDecimal;  
  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
import java.util.concurrent.TimeUnit;  
  
import mx.mbi.dgc.dao.negocio.DAOCincoVar;  
import mx.mbi.dgc.dominio.Evaluacion;  
import mx.mbi.dgc.jpa.entity.CatMesDiaFirma;  
import mx.mbi.dgc.jpa.entity.CatVariable;  
import mx.mbi.dgc.jpa.entity.TxCincominutalVar;  
  
import org.joda.time.DateTime;  
import org.joda.time.format.DateTimeFormat;  
import org.joda.time.format.DateTimeFormatter;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.annotation.Scope;  
import org.springframework.stereotype.Component;  
  
@Component  
@Scope("prototype")  
public class TaskSaveDeteccion implements Runnable {  
  
 @Autowired  
 private DAOCincoVar daoCincoVar;  
  
 private CatVariable catVariable;  
 private TxCincominutalVar varPrimeraValida;  
 private List<TxCincominutalVar> listaVarPrimeraValida;  
 private List<TxCincominutalVar> listaTxCincominutalVar;  
 private static BigDecimal pivote = new BigDecimal("1"); //1 = Pivote  
 private static BigDecimal noDetectado = new BigDecimal("-1"); //-1 = No detectado  
 private static BigDecimal valido = new BigDecimal("27"); //27 = VALIDO;  
 private static Evaluacion evalExitosa;  
 private Integer delay;  
  
 public TaskSaveDeteccion() {  
 super();  
 }  
  
 static {  
 evalExitosa = new Evaluacion();  
 evalExitosa.setRegla(valido); //27 = válido  
 evalExitosa.setStatus(true);  
 }  
  
 @Override  
 public void run() {  
 Boolean existePivote = true;  
 TimeUnit unit = TimeUnit.SECONDS;  
 List<Map> tuplaPersistencia = new ArrayList<Map>();  
 Map mapPersistencia = new HashMap();  
 System.out.printf("Detectando variable %s\n", catVariable.getNombre());  
 try {  
 this.listaTxCincominutalVar =  
 daoCincoVar.getListaVars(noDetectado, this.catVariable.getVariableId(),  
 this.catVariable.getCatPlanta().getPlantaId());  
 if (this.listaTxCincominutalVar != null && this.listaTxCincominutalVar.size() > 0) {  
 varPrimeraValida = this.listaTxCincominutalVar.get(0);  
 existePivote = false;  
 try {  
 listaVarPrimeraValida =  
 daoCincoVar.obtenerUltimaValida(this.catVariable  
 .getCatPlanta()  
 .getPlantaId(),  
 this.catVariable.getVariableId());  
 for (TxCincominutalVar txCincominutalVar : listaVarPrimeraValida) {  
 if (txCincominutalVar.getTxCincominutal()  
 .getCatMesDiaFirma()  
 .getFechaFirma()  
 .before(varPrimeraValida.getTxCincominutal()  
 .getCatMesDiaFirma()  
 .getFechaFirma())) {  
 varPrimeraValida = txCincominutalVar;  
 existePivote = true;  
 }  
 }  
 } catch (Exception ex) {  
 varPrimeraValida = null;  
 }  
  
 tuplaPersistencia = new ArrayList<Map>();  
 mapPersistencia = new HashMap();  
  
 mapPersistencia.put("Evaluacion", evalExitosa);  
 mapPersistencia.put("TxCincominutalVar", varPrimeraValida);  
 tuplaPersistencia.add(mapPersistencia);  
  
 if (!existePivote) {  
 varPrimeraValida.setEstatusDn(pivote);  
 varPrimeraValida.setFechaUm(new Date());  
 varPrimeraValida.setFechaDn(varPrimeraValida.getFechaUm());  
 synchronized (daoCincoVar) {  
 daoCincoVar.save(varPrimeraValida);  
 }  
 }  
  
  
 int control = 0;  
 for (TxCincominutalVar txVar : this.listaTxCincominutalVar) {  
 if (!existePivote) {  
 existePivote = true;  
 continue;  
 }  
 Evaluacion eval;  
 if (txVar.getTxCincominutal()  
 .getCatMesDiaFirma()  
 .getCatFirmaCincominutal()  
 .getFirma()  
 .equals("00000005") && isPrimeraDiaDelMes(txVar.getTxCincominutal().getCatMesDiaFirma())) {  
 eval = validaIPM(txVar);  
 } else {  
 if (txVar.getTxCincominutal()  
 .getCatMesDiaFirma()  
 .getCatFirmaCincominutal()  
 .getFirma()  
 .equals("23550000") && isUltimoDiaDelMes(txVar.getTxCincominutal().getCatMesDiaFirma())) {  
 eval = validaUPM(txVar, varPrimeraValida);  
 } else {  
 eval = validaReglas(txVar, varPrimeraValida);  
 }  
 }  
 mapPersistencia = new HashMap();  
 mapPersistencia.put("Evaluacion", eval);  
 mapPersistencia.put("TxCincominutalVar", txVar);  
 tuplaPersistencia.add(mapPersistencia);  
 if (eval.isStatus()) {  
 varPrimeraValida.setEstatusDn(valido);  
 varPrimeraValida.setFechaUm(new Date());  
 varPrimeraValida.setFechaDn(varPrimeraValida.getFechaUm());  
 synchronized (daoCincoVar) {  
 daoCincoVar.save(varPrimeraValida);  
 }  
  
 varPrimeraValida = txVar;  
 List<TxCincominutalVar> listaInsertVar = new ArrayList<>();  
 for (Map mapVar : tuplaPersistencia) {  
 TxCincominutalVar txUpdateVar = ((TxCincominutalVar) mapVar.get("TxCincominutalVar"));  
 txUpdateVar.setEstatusDn(((Evaluacion) mapVar.get("Evaluacion")).getRegla());  
 txUpdateVar.setFechaUm(new Date());  
 txUpdateVar.setFechaDn(txUpdateVar.getFechaUm());  
 listaInsertVar.add(txUpdateVar);  
 }  
 tuplaPersistencia = new ArrayList<Map>();  
 mapPersistencia = new HashMap();  
 mapPersistencia.put("Evaluacion", evalExitosa);  
 mapPersistencia.put("TxCincominutalVar", varPrimeraValida);  
 tuplaPersistencia.add(mapPersistencia);  
  
 if (listaInsertVar != null && listaInsertVar.size() > 0) {  
 synchronized (daoCincoVar) {  
 daoCincoVar.saveAll(listaInsertVar);  
 }  
 }  
  
 varPrimeraValida.setEstatusDn(pivote);  
 varPrimeraValida.setFechaUm(new Date());  
 varPrimeraValida.setFechaDn(varPrimeraValida.getFechaUm());  
 synchronized (daoCincoVar) {  
 daoCincoVar.save(varPrimeraValida);  
 }  
 }  
 control++;  
 if (control % 1000 == 0) {  
 System.gc();  
 }  
 }  
  
 } else {  
 System.out.println("TaskSavwDeteccion.run: La lista de cincominutales var es vacia");  
 }  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 }  
  
 private Evaluacion validaUPM(TxCincominutalVar txCvar, TxCincominutalVar varPrimeraValida) {  
 double nuevoValor = txCvar.getValorFormato().doubleValue();  
 double valor = varPrimeraValida.getValorFormato().doubleValue() - txCvar.getValorFormato().doubleValue();  
 Evaluacion eval = new Evaluacion();  
 eval.setRegla(new BigDecimal(27)); //27 = válido  
 eval.setStatus(true);  
 //Validar congelamiento = 15  
 boolean quitarAddon = false;  
 if (valor == 0) {  
 nuevoValor = txCvar.getValorFormato().doubleValue() + 0.01;  
   
 eval.setStatus(true);  
 eval.setRegla(new BigDecimal(32)); //32 CONGELADA ULTIMA DEL MES  
 }  
  
 //Validar INVALIDO POR FUERA LIMITE INFERIOR = 17  
 if (txCvar.getValorFormato().doubleValue() < txCvar.getCatVariable()  
 .getValorMinimo()  
 .doubleValue()) {  
 nuevoValor = txCvar.getValorFormato().doubleValue();  
 eval.setStatus(false);  
 eval.setRegla(new BigDecimal(17));  
 return eval;  
 }  
 //Validar INVALIDO POR FUERA LIMITE SUPERIOR = 16  
 if (txCvar.getValorFormato().doubleValue() > txCvar.getCatVariable()  
 .getValorMaximo()  
 .doubleValue()) {  
 nuevoValor = txCvar.getValorFormato().doubleValue();  
 eval.setStatus(false);  
 eval.setRegla(new BigDecimal(16));  
 return eval;  
 }  
 txCvar.setValorFormato(new BigDecimal(nuevoValor));  
 return eval;  
 }  
  
 private Evaluacion validaIPM(TxCincominutalVar txCvar) {  
 //Valida limites inferior y superior  
 Evaluacion eval = new Evaluacion();  
 eval.setRegla(new BigDecimal(27)); //27 = válido  
 eval.setStatus(true);  
 //Validar INVALIDO POR FUERA LIMITE INFERIOR = 17  
 if (txCvar.getValorFormato().doubleValue() < txCvar.getCatVariable()  
 .getValorMinimo()  
 .doubleValue()) {  
 eval.setStatus(false);  
 eval.setRegla(new BigDecimal(17));  
 return eval;  
 }  
 //Validar INVALIDO POR FUERA LIMITE SUPERIOR = 16  
 if (txCvar.getValorFormato().doubleValue() > txCvar.getCatVariable()  
 .getValorMaximo()  
 .doubleValue()) {  
 eval.setStatus(false);  
 eval.setRegla(new BigDecimal(16));  
 return eval;  
 }  
 return eval;  
 }  
  
 private Evaluacion validaReglas(TxCincominutalVar varValidar, TxCincominutalVar varPrimeraValida) {  
 double valor = varPrimeraValida.getValorFormato().doubleValue() - varValidar.getValorFormato().doubleValue();  
 Evaluacion eval = new Evaluacion();  
 eval.setRegla(new BigDecimal(27)); //27 = válido  
 eval.setStatus(true);  
 //Validar congelamiento = 15  
 if (valor == 0) {  
 eval.setStatus(false);  
 eval.setRegla(new BigDecimal(15));  
 return eval;  
 }  
 //Validar INVALIDO POR DIFERENCIA MAXIMA = 19  
 if ((valor + varValidar.getCatVariable()  
 .getDiferenciaMaxima()  
 .doubleValue()) < 0) {  
 eval.setStatus(false);  
 eval.setRegla(new BigDecimal(19));  
 return eval;  
 }  
 //Validar INVALIDO POR DIFERENCIA MAXIMA = 18  
 if ((valor - varValidar.getCatVariable()  
 .getDiferenciaMaxima()  
 .doubleValue()) > 0) {  
 eval.setStatus(false);  
 eval.setRegla(new BigDecimal(18));  
 return eval;  
 }  
 //Validar INVALIDO POR FUERA LIMITE INFERIOR = 17  
 if (varValidar.getValorFormato().doubleValue() < varValidar.getCatVariable()  
 .getValorMinimo()  
 .doubleValue()) {  
 eval.setStatus(false);  
 eval.setRegla(new BigDecimal(17));  
 return eval;  
 }  
 //Validar INVALIDO POR FUERA LIMITE SUPERIOR = 16  
 if (varValidar.getValorFormato().doubleValue() > varValidar.getCatVariable()  
 .getValorMaximo()  
 .doubleValue()) {  
 eval.setStatus(false);  
 eval.setRegla(new BigDecimal(16));  
 return eval;  
 }  
 return eval;  
 }  
   
 private boolean isUltimoDiaDelMes(CatMesDiaFirma inicioTemporal){  
 boolean retorno = false;  
 String pattern = "yyyy\_MMdd";  
 String cadenaTiempo = inicioTemporal.getCatMesDia().getCatAnioMes().getAnioMes() + inicioTemporal.getCatMesDia().getDia();  
 DateTimeFormatter builderDateTimeFormat = DateTimeFormat.forPattern(pattern);  
 DateTime inicioTemporalDateTime = DateTime.parse(cadenaTiempo, builderDateTimeFormat);  
 int dia = inicioTemporalDateTime.getDayOfMonth();  
 int finMes = inicioTemporalDateTime.dayOfMonth().getMaximumValue();  
 if(dia == finMes){  
 retorno = true;  
 }  
 return retorno;  
 }  
   
 private boolean isPrimeraDiaDelMes(CatMesDiaFirma inicioTemporal){  
 boolean retorno = false;  
 String pattern = "yyyy\_MMdd";  
 String cadenaTiempo =inicioTemporal.getCatMesDia().getCatAnioMes().getAnioMes() + inicioTemporal.getCatMesDia().getDia();  
 DateTimeFormatter builderDateTimeFormat = DateTimeFormat.forPattern(pattern);  
 DateTime inicioTemporalDateTime = DateTime.parse(cadenaTiempo, builderDateTimeFormat);  
 int dia = inicioTemporalDateTime.getDayOfMonth();  
 if(dia == 1){  
 retorno = true;  
 }  
 return retorno;  
 }  
  
  
 public CatVariable getCatVariable() {  
 return catVariable;  
 }  
  
 public void setCatVariable(CatVariable catVariable) {  
 this.catVariable = catVariable;  
 }  
  
 public TxCincominutalVar getVarPrimeraValida() {  
 return varPrimeraValida;  
 }  
  
 public void setVarPrimeraValida(TxCincominutalVar varPrimeraValida) {  
 this.varPrimeraValida = varPrimeraValida;  
 }  
  
 public List<TxCincominutalVar> getListaTxCincominutalVar() {  
 return listaTxCincominutalVar;  
 }  
  
 public void setListaTxCincominutalVar(List<TxCincominutalVar> listaTxCincominutalVar) {  
 this.listaTxCincominutalVar = listaTxCincominutalVar;  
 }  
  
 public Integer getDelay() {  
 return delay;  
 }  
  
 public void setDelay(Integer delay) {  
 this.delay = delay;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\BusinessFacturacion.java

package mx.mbi.dgc.business.facturacion;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatFacturaProductos;  
import mx.mbi.dgc.dao.DAOCatPlantaTipoFacturas;  
import mx.mbi.dgc.dao.negocio.facturacion.CustomDAOCatCompaniaPlantas;  
import mx.mbi.dgc.dao.negocio.facturacion.CustomDAOCatServicioImpuestos;  
import mx.mbi.dgc.dao.negocio.facturacion.CustomDAOTxDocumentos;  
import mx.mbi.dgc.dominio.DominioCatCompaniaPlantas;  
import mx.mbi.dgc.dominio.DominioCatFacturaProductos;  
import mx.mbi.dgc.dominio.DominioCatPlanta;  
import mx.mbi.dgc.dominio.DominioCatPlantaTipoFacturas;  
import mx.mbi.dgc.dominio.DominioCatServicioImpuestos;  
import mx.mbi.dgc.dominio.DominioCatServicios;  
import mx.mbi.dgc.dominio.DominioTxDocumentos;  
import mx.mbi.dgc.dominio.negocio.facturacion.DominioFacturacion;  
import mx.mbi.dgc.jpa.entity.CatCompania;  
import mx.mbi.dgc.jpa.entity.CatCompaniaPlantas;  
import mx.mbi.dgc.jpa.entity.CatFacturaProductos;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.jpa.entity.CatPlantaTipoFacturas;  
import mx.mbi.dgc.jpa.entity.CatServicioImpuestos;  
import mx.mbi.dgc.jpa.entity.CatServicios;  
import mx.mbi.dgc.jpa.entity.TxDocumentos;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessFacturacion implements IfaceBusinessFacturacion {  
 @Autowired  
 private CustomDAOTxDocumentos daoTxDocumentos;  
  
 @Autowired  
 private CustomDAOCatCompaniaPlantas daoCatCompanaPlantas;  
  
 @Autowired  
 private CustomDAOCatServicioImpuestos daoCatServicioImpuestos;  
  
 @Autowired  
 private DAOCatPlantaTipoFacturas daoCatPlantaTipoFacturas;  
  
 @Autowired  
 private DAOCatFacturaProductos daoCatFacturaProductos;  
  
 @Override  
 public int getMaxFolioBySerie(Object obj) throws Exception {  
 try {  
 DominioTxDocumentos dom = (DominioTxDocumentos) obj;  
 TxDocumentos entity = new TxDocumentos();  
 JRUtil.copyProperties(entity, dom);  
 return daoTxDocumentos.getMaxFolioBySerie(entity.getSerie());  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessTxDocumentos.getMaxFolioBySerie: " + ex);  
 }  
 }  
  
 public DominioCatCompaniaPlantas findCompaniaPlantaByPlantaId(Object obj) throws Exception {  
 try {  
 DominioFacturacion domFacturacion = (DominioFacturacion) obj;  
 CatPlanta catPlanta = new CatPlanta();  
 CatCompania catCompania = new CatCompania();  
 JRUtil.copyProperties(catPlanta, domFacturacion.getCatPlanta());  
 JRUtil.copyProperties(catCompania, domFacturacion.getCatCompania());  
 DominioCatCompaniaPlantas dom = new DominioCatCompaniaPlantas();  
 CatCompaniaPlantas entity = daoCatCompanaPlantas.findByCatPlantaAndCatCompania(catPlanta, catCompania);  
 JRUtil.copyProperties(dom, entity);  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessFacturacion.findCompaniaPlantaByPlantaId: " + ex);  
 }  
 }  
  
 public List<DominioCatServicioImpuestos> findImpuestosByServicio(Object obj) throws Exception {  
 try {  
 DominioCatServicios domSer = (DominioCatServicios) obj;  
 CatServicios catServicios = new CatServicios();  
 JRUtil.copyProperties(catServicios, domSer);  
  
 List<DominioCatServicioImpuestos> lista = new ArrayList<>();  
 List<CatServicioImpuestos> objs = daoCatServicioImpuestos.findByCatServicios(catServicios);  
 for (CatServicioImpuestos entity : objs) {  
 DominioCatServicioImpuestos dom = new DominioCatServicioImpuestos();  
 JRUtil.copyProperties(dom, entity);  
 System.out.println("retenido en entity: " + entity.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getRetenido());  
 System.out.println("retenido en dom convertido: " + dom.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getRetenido());  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatPlantaTipoFacturas.findAll: " + ex);  
 }  
 }  
  
 public List<DominioCatPlantaTipoFacturas> findTipoFacturasByCatPlanta(Object obj) throws Exception {  
 try {  
 DominioCatPlanta domPlanta = (DominioCatPlanta) obj;  
 CatPlanta catPlanta = new CatPlanta();  
 JRUtil.copyProperties(catPlanta, domPlanta);  
  
 List<DominioCatPlantaTipoFacturas> lista = new ArrayList<>();  
 List<CatPlantaTipoFacturas> objs = daoCatPlantaTipoFacturas.findByCatPlanta(catPlanta);  
 for (CatPlantaTipoFacturas entity : objs) {  
 DominioCatPlantaTipoFacturas dom = new DominioCatPlantaTipoFacturas();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatPlantaTipoFacturas.findAll: " + ex);  
 }  
 }  
  
 public List<DominioCatFacturaProductos> findFacturaProductosByTxDocumento(Object obj) throws Exception {  
 try {  
 DominioTxDocumentos domTxDocumento = (DominioTxDocumentos) obj;  
 TxDocumentos txDocumento = new TxDocumentos();  
 JRUtil.copyProperties(txDocumento, domTxDocumento);  
  
 List<DominioCatFacturaProductos> lista = new ArrayList<>();  
 List<CatFacturaProductos> objs = daoCatFacturaProductos.findByTxDocumentos(txDocumento);  
 for (CatFacturaProductos entity : objs) {  
 DominioCatFacturaProductos dom = new DominioCatFacturaProductos();  
 JRUtil.copyProperties(dom, entity);  
 lista.add(dom);  
 }  
 return lista;  
 } catch (Exception ex) {  
 throw new Exception("Error in BusinessCatPlantaTipoFacturas.findAll: " + ex);  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\BusinessFacturacionSelladoTimbrado.java

package mx.mbi.dgc.business.facturacion;  
  
import java.io.BufferedReader;  
import java.io.BufferedWriter;  
import java.io.File;  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.io.InputStreamReader;  
import java.io.PrintWriter;  
import java.io.StringWriter;  
  
import java.nio.charset.StandardCharsets;  
  
import java.text.DecimalFormat;  
  
import java.util.List;  
import java.util.Set;  
  
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;  
import javax.xml.parsers.ParserConfigurationException;  
import javax.xml.transform.OutputKeys;  
import javax.xml.transform.Transformer;  
import javax.xml.transform.TransformerException;  
import javax.xml.transform.TransformerFactory;  
import javax.xml.transform.dom.DOMSource;  
import javax.xml.transform.stream.StreamResult;  
  
import mx.mbi.dgc.business.dyc.BusinessDeteccionNorma;  
import mx.mbi.dgc.client.facturacion.Response;  
import mx.mbi.dgc.client.facturacion.WsTimbrado;  
import mx.mbi.dgc.dominio.DominioCatClientes;  
import mx.mbi.dgc.dominio.DominioCatDatosFiscales;  
import mx.mbi.dgc.dominio.DominioCatFacturaProductos;  
import mx.mbi.dgc.dominio.DominioCatServicioImpuestos;  
import mx.mbi.dgc.dominio.DominioTxDocumentos;  
import mx.mbi.dgc.dominio.DominioTxFacturaUuids;  
import mx.mbi.dgc.global.Globales;  
import mx.mbi.dgc.jpa.entity.CatDatosFiscales;  
import mx.mbi.dgc.jpa.entity.CatFacturaProductos;  
import mx.mbi.dgc.jpa.entity.CatServicioImpuestos;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.joda.time.DateTime;  
import org.joda.time.format.DateTimeFormat;  
import org.joda.time.format.DateTimeFormatter;  
  
import org.springframework.stereotype.Service;  
  
import org.w3c.dom.DOMImplementation;  
import org.w3c.dom.Document;  
import org.w3c.dom.Element;  
  
@Service  
public class BusinessFacturacionSelladoTimbrado implements IfaceFacturacionSelladoTimbrado {  
 private static final Logger LOG = LogManager.getLogger(BusinessDeteccionNorma.class);  
  
 WsTimbrado wsTimbrado;  
  
 public BusinessFacturacionSelladoTimbrado() {  
 try {  
 this.wsTimbrado = new WsTimbrado();  
 } catch (Exception e) {  
 System.out.println("Servicio de timbrado caído");  
 }  
 }  
  
  
 private String leerArchivo(String fileNAme) {  
 String retorno = "";  
 File f = new File(fileNAme);  
 if (!f.exists() && !f.isDirectory()) {  
 return null;  
 }  
 FileReader fr = null;  
  
 BufferedReader br = null;  
  
 try {  
 fr = new FileReader(f);  
 br = new BufferedReader(fr);  
 String linea;  
 while ((linea = br.readLine()) != null)  
 retorno += linea;  
 } catch (Exception e) {  
 e.printStackTrace();  
 } finally {  
 try {  
 if (null != fr) {  
 fr.close();  
 }  
 } catch (Exception e2) {  
 e2.printStackTrace();  
 }  
 }  
  
 return retorno;  
 }  
  
 private boolean escribirArchivo(String fileName, String texto) {  
 boolean retorno = true;  
 File f = new File(fileName);  
  
 //Escritura  
 try {  
 FileWriter w = new FileWriter(f);  
 BufferedWriter bw = new BufferedWriter(w);  
 PrintWriter wr = new PrintWriter(bw);  
 wr.write(texto);  
 wr.close();  
 bw.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 return false;  
 }  
 return retorno;  
 }  
  
 public String cfdiCadenaOriginal(DominioTxDocumentos txDocumentos) {  
  
 String cadenaOriginalAux = "||";  
  
 DecimalFormat dfImpuesto = new DecimalFormat("0.000000");  
 DecimalFormat importeImpuesto = new DecimalFormat("0.00");  
 //DominioComprobante comprobante = (DominioComprobante) txDocumentos.get("comprobante");  
  
 cadenaOriginalAux += txDocumentos.getCatVersion().getVersion() + "|";  
 if (txDocumentos.getSerie() != null) {  
 cadenaOriginalAux += txDocumentos.getSerie() + "|";  
 }  
 if (txDocumentos.getFolio() != null) {  
 cadenaOriginalAux += txDocumentos.getFolio() + "|";  
 }  
  
 DateTimeFormatter builderFecha = DateTimeFormat.forPattern("yyyy-MM-dd");  
 DateTimeFormatter builderHoraMinuto = DateTimeFormat.forPattern("HH:mm:ss");  
 DateTime fecha = new DateTime(txDocumentos.getFechaHoraCreacion());  
  
 DecimalFormat myFormatter = new DecimalFormat("0.00");  
 cadenaOriginalAux += builderFecha.print(fecha) + "T" + builderHoraMinuto.print(fecha) + "|";  
 if (txDocumentos.getCatFormasPago() != null) {  
 cadenaOriginalAux += txDocumentos.getCatFormasPago().getNombre() + "|";  
 }  
 cadenaOriginalAux += txDocumentos.getNoCertificado() + "|";  
 if (txDocumentos.getCondicionesPago() != null) {  
 cadenaOriginalAux += txDocumentos.getCondicionesPago() + "|";  
 }  
 double subTotal = calculaSubtotalL(txDocumentos.getCatFacturaProductoses());  
 double descuentos = calculaDescuentoL(txDocumentos.getCatFacturaProductoses());  
 cadenaOriginalAux += subTotal + "|";  
 if (txDocumentos.getCatFacturaProductoses() != null && txDocumentos.getCatFacturaProductoses().size() > 0  
 /\*&&  
 txDocumentos.get("tipoCFDI") != "E"\*/) {  
 if (descuentos > 0)  
 cadenaOriginalAux += myFormatter.format(descuentos) + "|";  
 }  
 cadenaOriginalAux += txDocumentos.getCatMonedas().getClave() + "|";  
 if (txDocumentos.getCatMonedas().getTipoCambio() != null) {  
 cadenaOriginalAux += "1|";  
 }  
 cadenaOriginalAux +=  
 myFormatter.format(calculaTotalL(subTotal, descuentos, txDocumentos.getCatFacturaProductoses())) + "|";  
 cadenaOriginalAux += txDocumentos.getCatTipoDocumento().getExtension() + "|";  
 if (txDocumentos.getCatMetodoPago() != null) {  
 cadenaOriginalAux += txDocumentos.getCatMetodoPago().getNombre() + "|";  
 }  
 cadenaOriginalAux += txDocumentos.getCatCompaniaPlantas()  
 .getCatDatosFiscales()  
 .getCp() + "|";  
 /\*fjvh  
 if (txDocumentos.getConfirmacionSat() != null) {  
 cadenaOriginalAux += txDocumentos.getConfirmacionSat() + "|";  
 }  
 \*/  
  
 //CfdiRelacionados  
 if (txDocumentos.getCatTipoRelacion() != null && !txDocumentos.getCatTipoRelacion()  
 .getNombre()  
 .equals("No Aplica")) {  
 cadenaOriginalAux += txDocumentos.getCatTipoRelacion().getNombre() + "|";  
 for (DominioTxFacturaUuids txFacturaUuids : txDocumentos.getTxFacturaUuidsesForDocumentoAsociadoId()) {  
 cadenaOriginalAux += txFacturaUuids.getTxDocumentosByDocumentoAsociadoId().getUuidSat() + "|";  
 }  
 }  
  
 //Emisor  
 DominioCatDatosFiscales emisor = txDocumentos.getCatCompaniaPlantas().getCatDatosFiscales();  
 cadenaOriginalAux += txDocumentos.getCatCompaniaPlantas()  
 .getCatCompania()  
 .getRfc() + "|";  
 if (txDocumentos.getCatCompaniaPlantas()  
 .getCatCompania()  
 .getRazonSocial() != null) {  
 cadenaOriginalAux += txDocumentos.getCatCompaniaPlantas()  
 .getCatCompania()  
 .getRazonSocial() + "|";  
 }  
 cadenaOriginalAux += emisor.getCatRegimenFiscal().getClaveNumero() + "|";  
  
 DominioCatClientes receptor = txDocumentos.getCatClientes();  
 cadenaOriginalAux += receptor.getRfc() + "|";  
 if (receptor.getRazonSocial() != null) {  
 cadenaOriginalAux += receptor.getNombreComercial() + "|";  
 }  
 //fjvh cadenaOriginalAux += calculaDatosFiscales(receptor.getCatDatosFiscales());  
 if (receptor.getCatDatosFiscales().getRegistroIdTributario() != null) {  
 cadenaOriginalAux += receptor.getCatDatosFiscales().getRegistroIdTributario() + "|";  
 }  
 cadenaOriginalAux += txDocumentos.getCatUsoCfdi().getClave() + "|";  
  
 List<DominioCatFacturaProductos> listaDominioConceptos = txDocumentos.getCatFacturaProductoses();  
 for (DominioCatFacturaProductos valueConceptos : listaDominioConceptos) {  
 cadenaOriginalAux += eliminaEspacios(valueConceptos.getCatServicios()  
 .getSatProductosClaves()  
 .getClave()) + "|";  
 /\*fjvh  
 if (valueConceptos.getCatServicios() != null) {  
 cadenaOriginalAux += valueConceptos.getCatServicios().getNumeroIdentificacion() + "|";  
 }  
\*/  
 cadenaOriginalAux += valueConceptos.getCantidad() + "|";  
 cadenaOriginalAux += valueConceptos.getCatServicios()  
 .getSatProductosUnidades()  
 .getClave() + "|";  
 if (valueConceptos.getCatServicios().getSatProductosUnidades() != null) {  
 cadenaOriginalAux += valueConceptos.getCatServicios()  
 .getSatProductosUnidades()  
 .getClave() + "|";  
 }  
  
 cadenaOriginalAux += eliminaEspacios(valueConceptos.getCatServicios().getDescripcion()) + "|";  
 cadenaOriginalAux += valueConceptos.getPrecioUnitario() + "|";  
 cadenaOriginalAux += myFormatter.format(valueConceptos.getImporte()) + "|";  
 if (valueConceptos.getDescuento() != null && txDocumentos.getCatTipoDocumento()  
 .getExtension()  
 .equals("I")) {  
 if (valueConceptos.getDescuento().doubleValue() > 0) {  
 cadenaOriginalAux +=  
 myFormatter.format(valueConceptos.getDescuento().doubleValue() \*  
 valueConceptos.getPrecioUnitario().doubleValue() \*  
 valueConceptos.getCantidad().doubleValue()) + "|";  
 }  
 }  
 }  
 //Impuestos  
 List<DominioCatFacturaProductos> listaDominioImpuestos = txDocumentos.getCatFacturaProductoses();  
  
 double totalImpuestosRetenidos = 0;  
 for (DominioCatFacturaProductos impuestos : listaDominioImpuestos) {  
 if (impuestos.getCatServicios().getCatServicioImpuestoses() != null) {  
 for (DominioCatServicioImpuestos valueRetencionesT :  
 impuestos.getCatServicios().getCatServicioImpuestoses()) {  
 if (valueRetencionesT.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getRetenido()) {  
 cadenaOriginalAux += valueRetencionesT.getCatImpuestos().getNombre() + "|";  
 cadenaOriginalAux += valueRetencionesT.getCatImpuestos().getPorcentaje() + "|";  
 }  
 }  
 }  
 }  
 totalImpuestosRetenidos = calculaImpuestosL(listaDominioImpuestos, "retenidos");  
 if (totalImpuestosRetenidos > 0)  
 cadenaOriginalAux += myFormatter.format(totalImpuestosRetenidos) + "|";  
 double totalImpuestosTrasladados = 0;  
 for (DominioCatFacturaProductos impuestos : listaDominioImpuestos) {  
 for (DominioCatServicioImpuestos valueTrasladosT :  
 impuestos.getCatServicios().getCatServicioImpuestoses()) {  
 if (valueTrasladosT.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getTrasladado()) {  
  
 cadenaOriginalAux += myFormatter.format(impuestos.getImporte()) + "|";  
 cadenaOriginalAux += valueTrasladosT.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getClaveSat() + "|";  
 cadenaOriginalAux += valueTrasladosT.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getCatTipoFactoresSat()  
 .getNombre() + "|";  
 cadenaOriginalAux += dfImpuesto.format(valueTrasladosT.getCatImpuestos().getPorcentaje()) + "|";  
 cadenaOriginalAux +=  
 myFormatter.format(valueTrasladosT.getCatImpuestos()  
 .getPorcentaje()  
 .doubleValue() \*  
 impuestos.getImporte().doubleValue()) + "|";  
 }  
 }  
 }  
 for (DominioCatFacturaProductos impuestos : listaDominioImpuestos) {  
 for (DominioCatServicioImpuestos valueTrasladosT :  
 impuestos.getCatServicios().getCatServicioImpuestoses()) {  
 if (valueTrasladosT.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getTrasladado()) {  
  
 cadenaOriginalAux += valueTrasladosT.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getClaveSat() + "|";  
 cadenaOriginalAux += valueTrasladosT.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getCatTipoFactoresSat()  
 .getNombre() + "|";  
 cadenaOriginalAux += dfImpuesto.format(valueTrasladosT.getCatImpuestos().getPorcentaje()) + "|";  
 cadenaOriginalAux +=  
 myFormatter.format(valueTrasladosT.getCatImpuestos()  
 .getPorcentaje()  
 .doubleValue() \*  
 impuestos.getImporte().doubleValue()) + "|";  
 }  
 }  
 }  
 totalImpuestosTrasladados = calculaImpuestosL(listaDominioImpuestos, "trasladados");  
 cadenaOriginalAux += myFormatter.format(totalImpuestosTrasladados) + "|";  
  
 cadenaOriginalAux += "|";  
 return cadenaOriginalAux;  
 }  
  
  
 private String calculaDatosFiscales(CatDatosFiscales catDatosFiscales) {  
 String retorno = "";  
 retorno +=  
 catDatosFiscales.getCalle() + " " + catDatosFiscales.getNumExt() + " " + catDatosFiscales.getNumInt() +  
 ", " + catDatosFiscales.getColonia() + " " + catDatosFiscales.getDelegacionOMunicipio() + " " +  
 catDatosFiscales.getCatEstados().getNombre() + " " + catDatosFiscales.getCp() + "|";  
  
 return retorno;  
 }  
  
 private String calculaDatosFiscalesL(DominioCatDatosFiscales catDatosFiscales) {  
 String retorno = "";  
 retorno +=  
 catDatosFiscales.getCalle() + " " + catDatosFiscales.getNumExt() + " " + catDatosFiscales.getNumInt() +  
 ", " + catDatosFiscales.getColonia() + " " + catDatosFiscales.getDelegacionOMunicipio() + " " +  
 catDatosFiscales.getCatEstados().getNombre() + " " + catDatosFiscales.getCp() + "|";  
  
 return retorno;  
 }  
  
 private Double calculaSubtotal(Set<CatFacturaProductos> catFacturaProductoses) {  
 Double retorno = 0d;  
 for (CatFacturaProductos catFacturaProductos : catFacturaProductoses) {  
 retorno += catFacturaProductos.getImporte().doubleValue();  
 }  
 return retorno;  
 }  
  
 private Double calculaSubtotalL(List<DominioCatFacturaProductos> catFacturaProductoses) {  
 Double retorno = 0d;  
 for (DominioCatFacturaProductos catFacturaProductos : catFacturaProductoses) {  
 retorno += catFacturaProductos.getImporte().doubleValue();  
 }  
 return retorno;  
 }  
  
 private Double calculaDescuento(Set<CatFacturaProductos> catFacturaProductoses) {  
 Double retorno = 0d;  
 for (CatFacturaProductos catFacturaProductos : catFacturaProductoses) {  
 retorno = catFacturaProductos.getImporte().doubleValue();  
 }  
 return retorno;  
 }  
  
 private Double calculaDescuentoL(List<DominioCatFacturaProductos> catFacturaProductoses) {  
 Double retorno = 0d;  
 for (DominioCatFacturaProductos catFacturaProductos : catFacturaProductoses) {  
 retorno +=  
 catFacturaProductos.getDescuento().doubleValue() \* catFacturaProductos.getCantidad().doubleValue() \*  
 catFacturaProductos.getPrecioUnitario().doubleValue();  
 }  
 return retorno;  
 }  
  
 private Double calculaImpuestos(Set<CatFacturaProductos> catFacturaProductoses, String tipo) {  
 double retorno = 0d;  
 double importeProducto = 0d;  
  
 for (CatFacturaProductos catFacturaProductos : catFacturaProductoses) {  
 importeProducto = catFacturaProductos.getImporte().doubleValue();  
 for (CatServicioImpuestos catServicioImpuesto :  
 catFacturaProductos.getCatServicios().getCatServicioImpuestoses()) {  
 if (tipo.equals("all") || tipo.equals("retenido")) {  
 if (catServicioImpuesto.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getRetenido()) {  
 retorno += importeProducto \* catServicioImpuesto.getCatImpuestos()  
 .getPorcentaje()  
 .doubleValue();  
 }  
 }  
  
 if (tipo.equals("all") || tipo.equals("trasladado")) {  
 if (catServicioImpuesto.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getTrasladado()) {  
 retorno += importeProducto \* catServicioImpuesto.getCatImpuestos()  
 .getPorcentaje()  
 .doubleValue();  
 }  
 }  
 }  
 }  
  
 return retorno;  
 }  
  
 private Double calculaImpuestosL(List<DominioCatFacturaProductos> catFacturaProductoses, String tipo) {  
 double retorno = 0d;  
 double importeProducto = 0d;  
  
 for (DominioCatFacturaProductos catFacturaProductos : catFacturaProductoses) {  
 importeProducto = catFacturaProductos.getImporte().doubleValue();  
 for (DominioCatServicioImpuestos catServicioImpuesto :  
 catFacturaProductos.getCatServicios().getCatServicioImpuestoses()) {  
 if (tipo.equals("all") || tipo.equals("retenidos")) {  
 if (Boolean.parseBoolean(String.valueOf(catServicioImpuesto.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getRetenido()))) {  
 retorno += importeProducto \* catServicioImpuesto.getCatImpuestos()  
 .getPorcentaje()  
 .doubleValue();  
 }  
 }  
  
 if (tipo.equals("all") || tipo.equals("trasladados")) {  
 if (Boolean.parseBoolean(String.valueOf(catServicioImpuesto.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getTrasladado()))) {  
 retorno += importeProducto \* catServicioImpuesto.getCatImpuestos()  
 .getPorcentaje()  
 .doubleValue();  
 }  
 }  
 }  
 }  
  
 return retorno;  
 }  
  
 private Double calculaTotal(Double subtotal, double descuento, Set<CatFacturaProductos> catFacturaProductoses) {  
 double retorno = 0d;  
 double totalImpuestos = calculaImpuestos(catFacturaProductoses, "all");  
 retorno = subtotal - descuento + totalImpuestos;  
 return retorno;  
 }  
  
 private Double calculaTotalL(Double subtotal, double descuento,  
 List<DominioCatFacturaProductos> catFacturaProductoses) {  
 double retorno = 0d;  
 double totalImpuestos = calculaImpuestosL(catFacturaProductoses, "all");  
 retorno = subtotal - descuento + totalImpuestos;  
 return retorno;  
 }  
  
 @Override  
 public String creaCertificado(String file\_certificado) {  
 String so = System.getProperty("os.name");  
 String s = "";  
 // String filePem = file\_certificado.replace(".cer", ".pem");  
 String fileCer = file\_certificado;  
 Process p;  
 String ret = "";  
 //Creamos el pem en caso de no existir  
 // File f = new File(filePem);  
 // if (!f.exists() && !f.isDirectory()) {  
 try {  
 if (!so.equals("Linux")) {  
 p = Runtime.getRuntime()  
 .exec("c:\\OpenSSL-Win64\\bin\\openssl x509 -inform DER -outform PEM -in " + fileCer + " -pubkey");  
 } else {  
 p = Runtime.getRuntime().exec("openssl x509 -inform DER -outform PEM -in " + fileCer + " -pubkey");  
 }  
 BufferedReader stdInput = new BufferedReader(new InputStreamReader(p.getInputStream()));  
  
 BufferedReader stdError = new BufferedReader(new InputStreamReader(p.getErrorStream()));  
  
  
 // Leemos la salida del comando  
 // System.out.println("Ésta es la salida standard del comando:\n");  
 while ((s = stdInput.readLine()) != null) {  
 ret += s;  
 }  
  
 // Leemos los errores si los hubiera  
 //System.out.println("Ésta es la salida standard de error del comando (si la hay):\n");  
 while ((s = stdError.readLine()) != null) {  
 System.out.println(s);  
 }  
  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 // }  
 //Validamos que exista el PEM  
  
 // f = new File(filePem);  
 // if (!f.exists() && !f.isDirectory()) {  
 // return "No existe el archivo PEM para el certificado";  
 // }  
  
 /\* String cert = System.getProperty("os.name")  
 .toUpperCase()  
 .contains("LINUX") ? leerArchivo(filePem) : ret;\*/  
 String cert = ret;  
 cert = cert.substring(cert.indexOf("-----BEGIN CERTIFICATE"), cert.length());  
 cert = cert.replaceAll("-----BEGIN CERTIFICATE-----", "");  
 cert = cert.replaceAll("-----END CERTIFICATE-----", "");  
 cert = cert.replaceAll("-----BEGIN PUBLIC KEY-----", "");  
 cert = cert.replaceAll("-----END PUBLIC KEY-----", "");  
 cert = cert.replaceAll("\n", "");  
 return cert;  
 }  
  
 private int generaNumeroAleatorio() {  
 int minimo = 10000001;  
 int maximo = 19999999;  
 int num = (int) Math.floor(Math.random() \* (maximo - minimo + 1) + (minimo));  
 return num;  
 }  
  
 @Override  
 public String creaSello(String cadenaOriginal, String keyFile, String password) {  
 String so = System.getProperty("os.name");  
 String s = "";  
 Process p = null;  
 //String diferenciador = sprintf("%u", ip2long($\_SERVER["REMOTE\_ADDR"]))."\_".rand(0,1000);  
 String diferenciador = "" + generaNumeroAleatorio();  
 String ruta\_llave =  
 System.getProperty("os.name")  
 .toUpperCase()  
 .contains("LINUX") ? Globales.RUTA\_CERTIFICADO\_LINUX :  
 Globales.RUTA\_CERTIFICADO\_WINDOWS;  
 ruta\_llave += keyFile;  
 String key = ruta\_llave + ".pem";  
 String pssw = password;  
 try {  
 if (!so.equals("Linux")) {  
 p = Runtime.getRuntime()  
 .exec("c:\\OpenSSL-Win64\\bin\\openssl pkcs8 -inform DER -in " + ruta\_llave + " -out " + key +  
 " -passin pass:" + pssw);  
 } else {  
 p = Runtime.getRuntime()  
 .exec("openssl pkcs8 -inform DER -in " + ruta\_llave + " -out " + key + " -passin pass:" + pssw);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 try {  
 Thread.sleep(200);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 File f = new File(key);  
 if (!f.exists() && !f.isDirectory()) {  
 return "Error al generar el pem...($comando)<br>";  
 }  
  
  
 //Creamos un archivo con la cadena Original en utf8  
  
 //cadenaUTF8 = utf8\_encode($this -> cadenaOriginal);  
 byte[] cadenaUTF8 = cadenaOriginal.getBytes(StandardCharsets.UTF\_8);  
 String RUTACACHE =  
 System.getProperty("os.name")  
 .toUpperCase()  
 .contains("LINUX") ? Globales.RUTA\_CERTIFICADO\_LINUX :  
 Globales.RUTA\_CERTIFICADO\_WINDOWS;  
 ;  
 String filename = RUTACACHE + "utf8\_" + diferenciador + ".txt";  
 String filename2 = RUTACACHE + "selloFinal\_" + diferenciador + ".txt";  
 String filename3 = RUTACACHE + "a.out";  
 System.out.println("cadena original:" + cadenaOriginal);  
  
 String cadenaUTF8Str = new String(cadenaUTF8, StandardCharsets.UTF\_8);  
 System.out.println("string bytes cadena original:" + cadenaUTF8Str);  
  
 boolean seEscribio = escribirArchivo(filename, cadenaUTF8Str);  
  
 String opensslPath = System.getProperty("os.name")  
 .toUpperCase()  
 .contains("LINUX") ? "openssl" : "c:\\OpenSSL-Win64\\bin\\openssl";  
 try {  
  
 System.out.println("Executando : " + "c:\\OpenSSL-Win64\\bin\\openssl dgst -sha256 -sign " + key + " " +  
 filename + " | c:\\OpenSSL-Win64\\bin\\openssl enc -base64 -A");  
 String[] cmdArray = new String[8];  
 cmdArray[0] = opensslPath;  
 cmdArray[1] = "dgst";  
 cmdArray[2] = "-sha256";  
 cmdArray[3] = "-out";  
 cmdArray[4] = RUTACACHE + "test.out";  
 cmdArray[5] = "-sign";  
 cmdArray[6] = key;  
 cmdArray[7] = filename;  
 p = Runtime.getRuntime().exec(cmdArray);  
 String salidaComando1 = "";  
  
 /\*  
 InputStream initialStream = p.getInputStream()  
 byte[] buffer = new byte[initialStream.available()];  
 initialStream.read(buffer);  
  
 File targetFile = new File("src/main/resources/targetFile.tmp");  
 OutputStream outStream = new FileOutputStream(targetFile);  
 outStream.write(buffer);  
 \*/  
  
 //StringWriter writer = new StringWriter();  
 //IOUtils.copy(new Base64InputStream(p.getInputStream()), writer);  
  
 //String base64decodedString = writer.toString();  
  
 /\* InputStream finput = p.getInputStream();  
 System.out.println("datos disponibles: "+finput.available());  
 byte[] isBytes = new byte[finput.available()];  
 finput.read(isBytes);  
 finput.close();  
 System.out.println("isBytes:"+isBytes.toString());  
 File targetFile = new File(RUTACACHE + "test.out");  
 OutputStream outStream = new FileOutputStream(targetFile);  
 outStream.write(isBytes);  
 outStream.close();  
  
 String strSello = Base64.encodeBase64String(isBytes);  
  
 System.out.println("sellado no codificado2: " + strSello);\*/  
 BufferedReader stdInput = new BufferedReader(new InputStreamReader(p.getInputStream()));  
 BufferedReader stdError = new BufferedReader(new InputStreamReader(p.getErrorStream()));  
  
 try {  
 // Leemos la salida del comando  
 System.out.println("Ésta es la salida standard del comando:\n");  
 while ((s = stdInput.readLine()) != null) {  
 salidaComando1 += s;  
 }  
 // Leemos los errores si los hubiera  
 System.out.println("Ésta es la salida standard de error del comando (si la hay):\n");  
 while ((s = stdError.readLine()) != null) {  
 System.out.println(s);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 System.out.println("sellado no codificado: " + salidaComando1);  
 // System.out.println("sellado no codificado2: " + strSello);  
 // escribirArchivo(filename3, strSello);  
  
 cmdArray = new String[6];  
 cmdArray[0] = opensslPath;  
 cmdArray[1] = "enc";  
 cmdArray[2] = "-base64";  
 cmdArray[3] = "-A";  
 cmdArray[4] = "-in";  
 cmdArray[5] = RUTACACHE + "test.out";  
 p = Runtime.getRuntime().exec(cmdArray);  
  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 /\*  
 f = new File(filename);  
 if (f.exists()) {  
 f.delete();  
 }\*/  
 BufferedReader stdInput = new BufferedReader(new InputStreamReader(p.getInputStream()));  
  
 BufferedReader stdError = new BufferedReader(new InputStreamReader(p.getErrorStream()));  
 String sello\_auxF = "";  
 try {  
 // Leemos la salida del comando  
 System.out.println("Ésta es la salida standard del comando:\n");  
 while ((s = stdInput.readLine()) != null) {  
 sello\_auxF += s;  
 System.out.println(s);  
 }  
 // Leemos los errores si los hubiera  
 System.out.println("Ésta es la salida standard de error del comando (si la hay):\n");  
 while ((s = stdError.readLine()) != null) {  
 System.out.println(s);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 System.out.println("sello base64:" + sello\_auxF);  
 String sello\_aux = leerArchivo(filename2);  
 //Eliminamos el archivo creado  
 f = new File(filename2);  
 if (f.exists()) {  
 f.delete();  
 }  
 f = new File(filename3);  
 if (f.exists()) {  
 f.delete();  
 }  
  
 return sello\_auxF;  
 }  
  
 @Override  
 public String timbra(byte[] str\_xml) {  
 System.out.println("public String timbra(byte[] str\_xml): " + str\_xml.toString());  
 String retorno = "";  
 Response resultado = this.wsTimbrado  
 .getWsTimbradoSoap12()  
 .generaTimbre(str\_xml);  
 retorno = resultado.getTimbre();  
 System.out.println("timbre:" + retorno);  
 System.out.println("error:" + resultado.getError());  
 return retorno;  
 }  
  
 private String eliminaEspacios(String cadena) {  
 cadena = cadena.replaceAll("/\\s+/", " ");  
 return cadena;  
 }  
  
  
 private String convertXMLToString(Document doc) {  
 try {  
 StringWriter sw = new StringWriter();  
 TransformerFactory tf = TransformerFactory.newInstance();  
 Transformer transformer = tf.newTransformer();  
 transformer.setOutputProperty(OutputKeys.OMIT\_XML\_DECLARATION, "no");  
 transformer.setOutputProperty(OutputKeys.METHOD, "xml");  
 transformer.setOutputProperty(OutputKeys.INDENT, "yes");  
 transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");  
  
 transformer.transform(new DOMSource(doc), new StreamResult(sw));  
 return sw.toString();  
 } catch (Exception ex) {  
 throw new RuntimeException("Error converting to String", ex);  
 }  
 }  
  
  
 @Override  
 public String cfdiXML(DominioTxDocumentos cfdi) {  
 Document xml = null;  
 DecimalFormat df = new DecimalFormat("0.00");  
 DecimalFormat dfImpuesto = new DecimalFormat("0.000000");  
 try {  
 DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();  
 DocumentBuilder builder;  
 builder = factory.newDocumentBuilder();  
 DOMImplementation implementation = builder.getDOMImplementation();  
 xml = implementation.createDocument(null, "cfdiXml", null);  
 xml.setXmlVersion("1.0");  
 Element comprobante = xml.createElement("cfdi:Comprobante");  
 comprobante.setAttribute("xmlns:cfdi", "http://www.sat.gob.mx/cfd/3");  
 comprobante.setAttribute("xmlns:xsi", "http://www.w3.org/2001/XMLSchema-instance");  
 comprobante.setAttribute("xsi:schemaLocation",  
 "http://www.sat.gob.mx/cfd/3 http://www.sat.gob.mx/sitio\_internet/cfd/3/cfdv33.xsd");  
 comprobante.setAttribute("Version", cfdi.getCatVersion().getVersion());  
 if (cfdi.getSerie() != null)  
 comprobante.setAttribute("Serie", cfdi.getSerie());  
 if (cfdi.getFolio() != null)  
 comprobante.setAttribute("Folio", cfdi.getFolio());  
 //2018-09-06T19:09:46  
 DateTimeFormatter builderFecha = DateTimeFormat.forPattern("yyyy-MM-dd");  
 DateTimeFormatter builderHoraMinuto = DateTimeFormat.forPattern("HH:mm:ss");  
 DateTime fecha = new DateTime(cfdi.getFechaHoraCreacion());  
 comprobante.setAttribute("Fecha", builderFecha.print(fecha) + "T" + builderHoraMinuto.print(fecha));  
 comprobante.setAttribute("Certificado", cfdi.getCertificado());  
 comprobante.setAttribute("Sello", cfdi.getSelloDigitalEmisor());  
 if (cfdi.getCatFormasPago() != null)  
 comprobante.setAttribute("FormaPago", cfdi.getCatFormasPago().getNombre());  
 if (cfdi.getCondicionesPago() != null)  
 comprobante.setAttribute("CondicionesDePago", cfdi.getCondicionesPago());  
  
 double subTotal = calculaSubtotalL(cfdi.getCatFacturaProductoses());  
 comprobante.setAttribute("SubTotal", "" + subTotal);  
  
 comprobante.setAttribute("TipoCambio", "1");  
  
  
 double descuentos = 0;  
 if ((cfdi.getCatFacturaProductoses() != null && cfdi.getCatTipoDocumento()  
 .getExtension()  
 .equals("I"))) {  
 descuentos = calculaDescuentoL(cfdi.getCatFacturaProductoses());  
 if (descuentos > 0)  
 comprobante.setAttribute("Descuento", "" + df.format(descuentos));  
 }  
 comprobante.setAttribute("Moneda", cfdi.getCatMonedas().getClave());  
 comprobante.setAttribute("NoCertificado", cfdi.getNoCertificado());  
 /\*if (cfdi.getCatMonedas().getTipoCambio() != null)  
 comprobante.setAttribute("TipoCambio", "" + cfdi.getCatMonedas().getTipoCambio());\*/  
 comprobante.setAttribute("Total",  
 "" +  
 df.format(calculaTotalL(subTotal, descuentos, cfdi.getCatFacturaProductoses())));  
 comprobante.setAttribute("TipoDeComprobante", cfdi.getCatTipoDocumento().getExtension());  
 if (cfdi.getCatMetodoPago() != null)  
 comprobante.setAttribute("MetodoPago", cfdi.getCatMetodoPago().getNombre());  
 comprobante.setAttribute("LugarExpedicion", cfdi.getCatCompaniaPlantas()  
 .getCatDatosFiscales()  
 .getCp()  
 .toString());  
 if (cfdi.getConfirmacionSat() != null)  
 comprobante.setAttribute("Confirmacion", cfdi.getConfirmacionSat());  
  
 if (!cfdi.getCatTipoRelacion()  
 .getNombre()  
 .equals("No Aplica")) {  
 Element cfdiRelacionados = xml.createElement("cfdi:CfdiRelacionados");  
 cfdiRelacionados.setAttribute("TipoRelacion", cfdi.getCatTipoRelacion().getNombre());  
 for (DominioTxFacturaUuids txFacturaUuids : cfdi.getTxFacturaUuidsesForDocumentoAsociadoId()) {  
 Element cfdiRelacionado = xml.createElement("cfdi:CfdiRelacionado");  
 cfdiRelacionado.setAttribute("UUID",  
 txFacturaUuids.getTxDocumentosByDocumentoAsociadoId().getUuidSat());  
 cfdiRelacionados.appendChild(cfdiRelacionado);  
 }  
 comprobante.appendChild(cfdiRelacionados);  
 }  
  
 Element emisor = xml.createElement("cfdi:Emisor");  
  
 emisor.setAttribute("Rfc", cfdi.getCatCompaniaPlantas()  
 .getCatCompania()  
 .getRfc());  
 if (cfdi.getCatCompaniaPlantas()  
 .getCatCompania()  
 .getRazonSocial() != null)  
 emisor.setAttribute("Nombre", cfdi.getCatCompaniaPlantas()  
 .getCatCompania()  
 .getRazonSocial());  
 emisor.setAttribute("RegimenFiscal", cfdi.getCatCompaniaPlantas()  
 .getCatDatosFiscales()  
 .getCatRegimenFiscal()  
 .getClaveNumero());  
 comprobante.appendChild(emisor);  
  
 Element receptor = xml.createElement("cfdi:Receptor");  
 receptor.setAttribute("Rfc", cfdi.getCatClientes().getRfc());  
 if (cfdi.getCatClientes().getNombreComercial() != null)  
 receptor.setAttribute("Nombre", cfdi.getCatClientes().getNombreComercial());  
 if (!cfdi.getCatClientes()  
 .getCatDatosFiscales()  
 .getCatEstados()  
 .getCatPaises()  
 .getClaveSat()  
 .equals("MEX"))  
 receptor.setAttribute("ResidenciaFiscal",  
 calculaDatosFiscalesL(cfdi.getCatClientes().getCatDatosFiscales()));  
 if (cfdi.getCatClientes()  
 .getCatDatosFiscales()  
 .getRegistroIdTributario() != null)  
 receptor.setAttribute("NumRegIdTrib", cfdi.getCatClientes()  
 .getCatDatosFiscales()  
 .getRegistroIdTributario());  
 receptor.setAttribute("UsoCFDI", cfdi.getCatUsoCfdi().getClave());  
  
 //Nodo cfdi:Conceptos  
 comprobante.appendChild(receptor);  
 Element conceptos = xml.createElement("cfdi:Conceptos");  
 List<DominioCatFacturaProductos> listaDominioConceptos = cfdi.getCatFacturaProductoses();  
 for (DominioCatFacturaProductos valueConceptos : listaDominioConceptos) {  
 Element concepto = xml.createElement("cfdi:Concepto");  
 concepto.setAttribute("Cantidad", valueConceptos.getCantidad().toString());  
 concepto.setAttribute("ClaveProdServ", valueConceptos.getCatServicios()  
 .getSatProductosClaves()  
 .getClave());  
 /\*if (valueConceptos.getCatServicios() != null) {  
 concepto.setAttribute("NoIdentificacion",  
 valueConceptos.getCatServicios().getNumeroIdentificacion());  
 }\*/  
  
 concepto.setAttribute("ClaveUnidad", valueConceptos.getCatServicios()  
 .getSatProductosUnidades()  
 .getClave());  
 concepto.setAttribute("Descripcion",  
 eliminaEspacios(valueConceptos.getCatServicios().getDescripcion()));  
 concepto.setAttribute("Importe", df.format(valueConceptos.getImporte()));  
 if (valueConceptos.getCatServicios().getSatProductosUnidades() != null) {  
 concepto.setAttribute("Unidad", valueConceptos.getCatServicios()  
 .getSatProductosUnidades()  
 .getClave());  
 }  
  
 concepto.setAttribute("ValorUnitario", valueConceptos.getPrecioUnitario().toString());  
  
 if (valueConceptos.getDescuento() != null && cfdi.getCatTipoDocumento()  
 .getExtension()  
 .equals("I")) {  
 if (valueConceptos.getDescuento() != null && valueConceptos.getDescuento().doubleValue() > 0)  
 concepto.setAttribute("Descuento",  
 df.format(valueConceptos.getDescuento().doubleValue() \*  
 valueConceptos.getPrecioUnitario().doubleValue() \*  
 valueConceptos.getCantidad().doubleValue()));  
 }  
  
 Element impuestos = xml.createElement("cfdi:Impuestos");  
 List<DominioCatServicioImpuestos> setImpuestos =  
 valueConceptos.getCatServicios().getCatServicioImpuestoses();  
 Element traslados = xml.createElement("cfdi:Traslados");  
 for (DominioCatServicioImpuestos valueTrasladosT : setImpuestos) {  
 if (valueTrasladosT.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getTrasladado()) {  
 Element traslado = xml.createElement("cfdi:Traslado");  
 //traslado.setAttribute("Base", valueTrasladosT.getCatImpuestos().);  
  
 traslado.setAttribute("Base", df.format(valueConceptos.getImporte()));  
 traslado.setAttribute("Impuesto", valueTrasladosT.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getClaveSat());  
 traslado.setAttribute("TipoFactor", valueTrasladosT.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getCatTipoFactoresSat()  
 .getNombre());  
 traslado.setAttribute("Importe",  
 "" +  
 df.format((valueConceptos.getImporte().doubleValue() \*  
 valueTrasladosT.getCatImpuestos()  
 .getPorcentaje()  
 .doubleValue())));  
 traslado.setAttribute("TasaOCuota",  
 dfImpuesto.format(valueTrasladosT.getCatImpuestos().getPorcentaje()));  
 //traslado.setAttribute("Importe", valueTrasladosT.getImporte);  
 traslados.appendChild(traslado);  
 }  
 impuestos.appendChild(traslados);  
 }  
 /\*Eliminamos Retenciones fjvh  
 Element retenciones = xml.createElement("cfdi:Retenciones");  
 for (DominioCatServicioImpuestos valueRetenciones : setImpuestos) {  
 if (valueRetenciones.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getRetenido()) {  
 Element retencion = xml.createElement("cfdi:Retenciones");  
 //retencion.setAttribute("Base", valueTrasladosR.getCatImpuestos().);  
 retencion.setAttribute("Impuesto", valueRetenciones.getCatImpuestos().getCatTipoImpuestos().getClaveSat());  
 retencion.setAttribute("TipoFactor", valueRetenciones.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getCatTipoFactoresSat()  
 .getNombre());  
 retencion.setAttribute("TasaOCuota", String.valueOf( valueRetenciones.getCatImpuestos().getPorcentaje()));  
 //retencion.setAttribute("Importe", valueRetenciones.getImporte);  
 retenciones.appendChild(retencion);  
  
 }  
 }  
  
 impuestos.appendChild(retenciones);  
 \*/  
 concepto.appendChild(impuestos);  
 conceptos.appendChild(concepto);  
 }  
 comprobante.appendChild(conceptos);  
  
  
 double totalImpuestosRetenidos = 0;  
 double totalImpuestosTrasladados = 0;  
 listaDominioConceptos = cfdi.getCatFacturaProductoses();  
 Element impuestos = xml.createElement("cfdi:Impuestos");  
 Element traslados = xml.createElement("cfdi:Traslados");  
  
 //Element retenciones = xml.createElement("cfdi:Retenciones");  
 for (DominioCatFacturaProductos valueConceptos : listaDominioConceptos) {  
 List<DominioCatServicioImpuestos> setImpuestos =  
 valueConceptos.getCatServicios().getCatServicioImpuestoses();  
 for (DominioCatServicioImpuestos valueTrasladosT : setImpuestos) {  
 if (valueTrasladosT.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getTrasladado()) {  
 Element traslado = xml.createElement("cfdi:Traslado");  
 traslado.setAttribute("Impuesto", valueTrasladosT.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getClaveSat());  
 traslado.setAttribute("TipoFactor", valueTrasladosT.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getCatTipoFactoresSat()  
 .getNombre());  
  
 //if(!empty(valueTrasladosT.TasaOCuota))  
 traslado.setAttribute("TasaOCuota",  
 dfImpuesto.format(valueTrasladosT.getCatImpuestos().getPorcentaje()));  
 if (valueConceptos.getImporte() != null)  
 traslado.setAttribute("Importe",  
 "" +  
 df.format((valueConceptos.getImporte().doubleValue() \*  
 valueTrasladosT.getCatImpuestos()  
 .getPorcentaje()  
 .doubleValue())));  
 traslados.appendChild(traslado);  
 totalImpuestosTrasladados +=  
 (valueConceptos.getImporte().doubleValue() \*  
 valueTrasladosT.getCatImpuestos()  
 .getPorcentaje()  
 .doubleValue());  
 }  
 }  
 double calculaImpuesto = 0;  
 /\* Eliminamos retenciones temporalmente fjvh  
 \* for (DominioCatServicioImpuestos valueRetenciones : setImpuestos) {  
 if (valueRetenciones.getCatImpuestos()  
 .getCatTipoImpuestos()  
 .getRetenido()) {  
 Element retencionT = xml.createElement("cfdi:Retencion");  
 retencionT = xml.createElement("cfdi:Retencion");  
 retencionT.setAttribute("Impuesto", valueRetenciones.getCatImpuestos().getCatTipoImpuestos().getClaveSat());  
  
 calculaImpuesto = (valueConceptos.getImporte().doubleValue() \* valueRetenciones.getCatImpuestos().getPorcentaje().doubleValue());  
 retencionT.setAttribute("Importe", "" + calculaImpuesto);  
 retenciones.appendChild(retencionT);  
 totalImpuestosRetenidos += calculaImpuesto;  
 }  
 }\*/  
 }  
  
 DecimalFormat myFormatter = new DecimalFormat("0.00");  
 if (totalImpuestosTrasladados > 0) {  
 impuestos.setAttribute("TotalImpuestosTrasladados", myFormatter.format(totalImpuestosTrasladados));  
 impuestos.appendChild(traslados);  
 }  
 /\*Eliminamos retenciones temporalmente fjvh  
 if (totalImpuestosRetenidos > 0) {  
 impuestos.setAttribute("TotalImpuestosRetenidos", myFormatter.format(totalImpuestosRetenidos));  
 impuestos.appendChild(retenciones);  
 }  
 \*/  
 comprobante.appendChild(impuestos);  
 //final del xml  
 xml.getDocumentElement().appendChild(comprobante);  
 // Source source = new DOMSource(xml);  
 // return source.toString();  
 } catch (ParserConfigurationException e) {  
 e.printStackTrace();  
 }  
 return convertDocumentToString(xml);  
 }  
  
 private static String convertDocumentToString(Document doc) {  
 TransformerFactory tf = TransformerFactory.newInstance();  
 Transformer transformer;  
 try {  
 transformer = tf.newTransformer();  
 // below code to remove XML declaration  
 // transformer.setOutputProperty(OutputKeys.OMIT\_XML\_DECLARATION, "yes");  
 StringWriter writer = new StringWriter();  
 transformer.transform(new DOMSource(doc), new StreamResult(writer));  
 String output = writer.getBuffer().toString();  
 output = output.replaceAll("</cfdiXml>", "");  
 output = output.replaceAll("<cfdiXml>", "");  
 System.out.println(output);  
 return output;  
  
 } catch (TransformerException e) {  
 e.printStackTrace();  
 }  
  
 return null;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\BusinessMontosFactura.java

package mx.mbi.dgc.business.facturacion;  
  
import java.math.BigDecimal;  
import java.math.RoundingMode;  
  
import java.time.YearMonth;  
  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.List;  
  
import mx.mbi.dgc.business.dyc.BusinessDeteccionNorma;  
import mx.mbi.dgc.dao.DAOCatPlanta;  
import mx.mbi.dgc.dao.DAOTxEscenario;  
import mx.mbi.dgc.dao.DAOTxEvento;  
import mx.mbi.dgc.dao.DAOTxValorFacturacion;  
import mx.mbi.dgc.dao.negocio.DAOAnioMes;  
import mx.mbi.dgc.dao.negocio.DaoNegocioTxCincominutal;  
import mx.mbi.dgc.dao.negocio.facturacion.DaoNegocioCatComponentes;  
import mx.mbi.dgc.dao.negocio.facturacion.DaoNegocioCatFormulaComponentes;  
import mx.mbi.dgc.dao.negocio.facturacion.DaoNegocioTxFormulas;  
import mx.mbi.dgc.dominio.negocio.facturacion.calculos.DominioCalculoCargoFijo;  
import mx.mbi.dgc.dominio.negocio.facturacion.calculos.DominioCalculosCargosFijos;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.global.Globales;  
import mx.mbi.dgc.global.Globales.Plantas;  
import mx.mbi.dgc.jpa.entity.TxEvento;  
import mx.mbi.dgc.jpa.entity.TxValorFActuracion;  
import mx.mbi.dgc.utils.DateUtil;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.joda.time.DateTime;  
import org.joda.time.Period;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessMontosFactura implements IfaceMontosFactura {  
 private static final Logger LOG = LogManager.getLogger(BusinessDeteccionNorma.class);  
  
 // @Autowired  
 // CalculadoraPCFC calculadoraPcfc;  
 @Autowired  
 DAOTxEvento daoTxEvento;  
  
 @Autowired  
 DAOTxValorFacturacion daoValorFacturacion;  
  
 @Autowired  
 DaoNegocioTxCincominutal daoNegocioTxCincominutal;  
  
 @Autowired  
 DAOAnioMes daoAnioMes;  
  
 @Autowired  
 DaoNegocioTxFormulas daoNegocioTxFormulas;  
  
 @Autowired  
 DAOTxEscenario daoTxEscenario;  
  
 @Autowired  
 DaoNegocioCatFormulaComponentes daoNegocioCatFormulaComponentes;  
  
 @Autowired  
 DAOCatPlanta daoCatPlanta;  
  
 @Autowired  
 DaoNegocioCatComponentes daoNegocioCatComponentes;  
  
 private List<BigDecimal> capacidadesPorEvento;  
  
  
 private BigDecimal obtenerPCFCm(BigDecimal cfcm, BigDecimal kc, BigDecimal faddm) {  
 return cfcm.multiply(kc).multiply(faddm);  
 }  
  
 private BigDecimal obtenerCFCam(BigDecimal cfcm, BigDecimal f) {  
 return cfcm.multiply(f.multiply(new BigDecimal(.8)).add(new BigDecimal(1)));  
 }  
  
  
 private BigDecimal obtenerHm(int mes, int año) {  
 return new BigDecimal(DateUtil.diasDelMes(mes, año)).multiply(new BigDecimal(24));  
 }  
  
 private BigDecimal obtenerYm(BigDecimal hm, List<BigDecimal> xm) {  
 double total = 0;  
 for (BigDecimal xmi : xm) {  
 total += xmi.doubleValue();  
 }  
 return hm.subtract(new BigDecimal(total));  
 }  
  
 private BigDecimal obtenerKc() {  
 try {  
 return daoValorFacturacion.findByVariableFacturacion\_Nombre("KC").getValor();  
 } catch (DyCNormaException e) {  
 e.printStackTrace();  
 return null;  
 }  
 }  
  
  
 private BigDecimal obtenerCfcm(Date fecha) {  
 try {  
 return daoValorFacturacion.findByVariableFacturacion\_NombreAndFecha("CFCm",  
 new Date(fecha.getYear(),  
 fecha.getMonth(), 1))  
 .getValor();  
 } catch (DyCNormaException e) {  
 e.printStackTrace();  
 return null;  
 }  
  
 }  
  
 private List<BigDecimal> obtenerCfcmL(int n, int año) {  
 List<BigDecimal> lista = new ArrayList<>();  
 for (int i = 0; i < n; i++) {  
 Date d = new Date(año, i, 1);  
 try {  
 TxValorFActuracion valorFacturacion =  
 daoValorFacturacion.findByVariableFacturacion\_NombreAndFecha("CFCm", d);  
 if (valorFacturacion != null) {  
 lista.add(daoValorFacturacion.findByVariableFacturacion\_NombreAndFecha("CFCm", d).getValor());  
  
 System.out.println("valor para mes " + i + " año:" + año + ":" +  
 daoValorFacturacion.findByVariableFacturacion\_NombreAndFecha("CFCm", d)  
 .getValor());  
 } else {  
 System.out.println("No hay dato en mes " + i + " año:" + año);  
 }  
 } catch (DyCNormaException e) {  
 e.printStackTrace();  
 return lista;  
 }  
 }  
 return lista;  
 }  
  
 private BigDecimal obtenerT0() {  
 try {  
 return daoValorFacturacion.findByVariableFacturacion\_Nombre("T0").getValor();  
 } catch (DyCNormaException e) {  
 e.printStackTrace();  
 return null;  
 }  
 }  
  
 private BigDecimal obtenerT1() {  
 try {  
 return daoValorFacturacion.findByVariableFacturacion\_Nombre("T1").getValor();  
 } catch (DyCNormaException e) {  
 e.printStackTrace();  
 return null;  
 }  
 }  
  
 private BigDecimal obtenerPdgm(Date fecha) {  
 try {  
 return daoValorFacturacion.findByVariableFacturacion\_NombreAndFecha("PDGm",  
 new Date(fecha.getYear(),  
 fecha.getMonth(), 1))  
 .getValor();  
 } catch (DyCNormaException e) {  
 e.printStackTrace();  
 return null;  
 }  
 }  
  
 private BigDecimal obtenerPdd(List<BigDecimal> fdedmL) {  
 BigDecimal sumaFdedm = new BigDecimal(0);  
 for (BigDecimal fdedm : fdedmL) {  
 sumaFdedm = sumaFdedm.add(fdedm);  
 }  
 return sumaFdedm.divide(new BigDecimal(12), 15, RoundingMode.HALF\_UP);  
 }  
  
 private BigDecimal obtenerPdd(Plantas planta, YearMonth mes) {  
  
 //return calculadoraPcfc.calcularPDDm(planta, mes);  
 return new BigDecimal(.974);  
 }  
  
 private BigDecimal obtenerFaddm(BigDecimal fadd, BigDecimal hm, BigDecimal ym, List<BigDecimal> xm,  
 List<BigDecimal> d, BigDecimal kc) {  
 BigDecimal a = fadd.divide(hm, 15, RoundingMode.HALF\_UP);  
  
 BigDecimal sumatoria = new BigDecimal(0);  
  
 int i = 0;  
 for (BigDecimal xmi : xm) {  
 sumatoria = sumatoria.add(xmi.multiply(d.get(i++).divide(kc)));  
 }  
 BigDecimal b = ym.add(sumatoria);  
 return a.multiply(b);  
 }  
  
  
 private BigDecimal obtenerF(List<BigDecimal> cfcmL, BigDecimal t0, BigDecimal t1) {  
 BigDecimal numerador = new BigDecimal(0);  
 BigDecimal denominador = new BigDecimal(0);  
 System.out.println("numero de cfcm:" + cfcmL.size());  
 int i = 1;  
 for (BigDecimal cfci : cfcmL) {  
 System.out.println("valor num:" + cfci.divide(t0.divide(new BigDecimal(1200), 15, RoundingMode.HALF\_UP)  
 .add(new BigDecimal(1))  
 .pow(i + 31), 15, RoundingMode.HALF\_UP));  
  
 System.out.println("valor den:" + cfci.divide(t1.divide(new BigDecimal(1200), 15, RoundingMode.HALF\_UP)  
 .add(new BigDecimal(1))  
 .pow(i + 31), 15, RoundingMode.HALF\_UP));  
  
 numerador = numerador.add(cfci.divide(t0.divide(new BigDecimal(1200), 15, RoundingMode.HALF\_UP)  
 .add(new BigDecimal(1))  
 .pow(i + 31), 15, RoundingMode.HALF\_UP));  
 denominador = denominador.add(cfci.divide(t1.divide(new BigDecimal(1200), 15, RoundingMode.HALF\_UP)  
 .add(new BigDecimal(1))  
 .pow(i + 31), 15, RoundingMode.HALF\_UP));  
  
 i++;  
  
 }  
 System.out.println(numerador.doubleValue());  
 System.out.println(denominador.doubleValue());  
 return numerador.divide(denominador, 15, RoundingMode.HALF\_UP);  
 }  
  
 private BigDecimal obtenerVmin(BigDecimal pdgm) {  
 return pdgm.multiply(new BigDecimal(.4924));  
 }  
  
 private BigDecimal obtenerFcor(BigDecimal pdgm) {  
 return new BigDecimal(1.97).divide(pdgm, 15, RoundingMode.HALF\_UP);  
 }  
  
 private BigDecimal obtenerFadd(BigDecimal vmin, BigDecimal pdd, BigDecimal pdgm, BigDecimal fcor) {  
 if (capacidadesPorEvento.size() == 0) {  
 return new BigDecimal(1);  
 } else {  
 if (pdgm.doubleValue() <= .96) {  
 if (pdd.doubleValue() >= 0 && pdd.doubleValue() <= vmin.doubleValue())  
 return new BigDecimal(0);  
 if (pdd.doubleValue() > vmin.doubleValue() && pdd.doubleValue() < pdgm.doubleValue())  
 return fcor.multiply(pdd).subtract(new BigDecimal(.97));  
 if (pdd.doubleValue() >= pdgm.doubleValue() && pdd.doubleValue() <= .96)  
 return new BigDecimal(1);  
 if (pdd.doubleValue() > .96 && pdd.doubleValue() < 1)  
 return pdd.multiply(new BigDecimal(1.5)).subtract(new BigDecimal(.44));  
 } else {  
  
 if (pdd.doubleValue() >= 0 && pdd.doubleValue() <= vmin.doubleValue())  
 return new BigDecimal(0);  
 if (pdd.doubleValue() > vmin.doubleValue() && pdd.doubleValue() < pdgm.doubleValue())  
 return fcor.multiply(pdd).subtract(new BigDecimal(.97));  
 if (pdd.doubleValue() > .96 && pdd.doubleValue() < 1)  
 return pdd.multiply(new BigDecimal(1.5)).subtract(new BigDecimal(.44));  
 }  
 }  
 return null;  
 }  
  
 private List<BigDecimal> obtenerD() {  
 return capacidadesPorEvento;  
 }  
  
 private List<BigDecimal> obtenerXm() {  
 List<TxEvento> eventosFalla = daoTxEvento.findByClasificacionEventoId\_ClasificacionEventoId(new BigDecimal(2));  
 capacidadesPorEvento = new ArrayList<>();  
 List<BigDecimal> xms = new ArrayList<>();  
 for (TxEvento eventoFalla : eventosFalla) {  
 Period horas =  
 new Period(new DateTime(eventoFalla.getFechaInicio()), new DateTime(eventoFalla.getFechaFinal()));  
 double horasTotales = horas.getHours() + (horas.getMinutes() / 60) + (horas.getSeconds() / 3600);  
 xms.add(new BigDecimal(horasTotales));  
 capacidadesPorEvento.add(new BigDecimal(eventoFalla.getPotencia()));  
 }  
 return xms;  
 }  
  
 private BigDecimal obtenerCFCam(BigDecimal cprciom, BigDecimal cpdam, BigDecimal cidam) {  
  
 return cprciom.add(cpdam.add(cidam));  
 }  
  
 private BigDecimal obtenerCprciom(Date fecha) {  
 try {  
  
 return daoValorFacturacion.findByVariableFacturacion\_NombreAndFecha("CPRCIOm",  
 new Date(fecha.getYear(),  
 fecha.getMonth(), 1))  
 .getValor();  
 } catch (DyCNormaException e) {  
 e.printStackTrace();  
 }  
 return null;  
 }  
  
 private BigDecimal obtenerF() {  
 try {  
 return daoValorFacturacion.findByVariableFacturacion\_Nombre("f(CPDam)").getValor();  
 } catch (DyCNormaException e) {  
 e.printStackTrace();  
  
 }  
 return null;  
 }  
  
 private BigDecimal obtenerCpdm(Date fecha) {  
 try {  
 return daoValorFacturacion.findByVariableFacturacion\_NombreAndFecha("CPDm",  
 new Date(fecha.getYear(),  
 fecha.getMonth(), 1))  
 .getValor();  
 } catch (DyCNormaException e) {  
 e.printStackTrace();  
  
 }  
 return null;  
 }  
  
 private BigDecimal obtenerCpdam(BigDecimal f, BigDecimal cpdm, BigDecimal t1, BigDecimal t0) {  
 return new BigDecimal(1).subtract(f)  
 .multiply(cpdm)  
 .add(t1.divide(t0, 15, RoundingMode.HALF\_UP).multiply(f.multiply(cpdm)));  
 }  
  
 private BigDecimal obtenerCidm(Date fecha) {  
 try {  
 return daoValorFacturacion.findByVariableFacturacion\_NombreAndFecha("CIDm",  
 new Date(fecha.getYear(),  
 fecha.getMonth(), 1))  
 .getValor();  
 } catch (DyCNormaException e) {  
 e.printStackTrace();  
  
 }  
 return null;  
 }  
  
 private BigDecimal obtenerCidam(BigDecimal cpdam, BigDecimal t1, BigDecimal t0, BigDecimal cpdm, BigDecimal cidm) {  
 return t1.divide(t0, 15, RoundingMode.HALF\_UP).multiply(cpdam.divide(cpdm).multiply(cidm));  
 }  
/\*  
 private List<BigDecimal> obtenerFdedmL(Date fecha) {  
 try {  
  
 Calendar cal = Calendar.getInstance();  
 List<BigDecimal> fdedmL = new ArrayList<>();  
 for (int i = 0; i < 12; i++) {  
 fdedmL.add(daoValorFacturacion.findByVariableFacturacion\_NombreAndFecha("fdedm",  
 new Date(fecha.getYear(),  
 fecha.getMonth(), 1))  
 .getValor());  
  
 cal.setTime(fecha);  
 cal.add(Calendar.MONTH, -1);  
 fecha = cal.getTime();  
 }  
  
 return fdedmL;  
  
 } catch (DyCNormaException e) {  
 e.printStackTrace();  
 }  
 return null;  
 }  
\*/  
 @Override  
 public DominioCalculosCargosFijos montosPCFC(DominioCalculosCargosFijos dom) {  
  
 int n =  
 dom.getFechaFactura().getMonth() +  
 1; //El último mes para el cual se hubieran cotizado Cargos Fijos de Capacidad, de acuerdo con lo establecido en la propuesta  
 BigDecimal cfcam = new BigDecimal(0);  
 BigDecimal cfcm = new BigDecimal(0);  
  
 BigDecimal t0 =  
 obtenerT0(); //Rendimiento hasta el vencimiento en los Bonos de Tesoreria una fecha anterior a la fecha de presentación de la Propuesta.  
 BigDecimal t1 =  
 obtenerT1(); //Rendimiento hasta el vencimiento en los Bonos de Tesoreria a una fecha de Inicio.  
 if (dom.getCatPlanta()  
 .getPlantaId()  
 .intValue() == Globales.EAT) {  
 List<BigDecimal> cfcL =  
 obtenerCfcmL(n, dom.getFechaFactura().getYear()); //Lista de Cargo fijo de capacidad por dia en el mes  
  
 BigDecimal f = obtenerF(cfcL, t0, t1); //  
  
 cfcm = obtenerCfcm(dom.getFechaFactura()); //Cargo Fijo de Capacidad para el mes m usd/kWm  
  
 cfcam = obtenerCFCam(cfcm, f); //Cargo Fijo de Capacidad para el mes m ajuwstado usd/kWm  
 } else {  
 if (dom.getCatPlanta()  
 .getPlantaId()  
 .intValue() == Globales.EST) {  
 BigDecimal cprciom = obtenerCprciom(dom.getFechaFactura());  
 BigDecimal f = obtenerF();  
 BigDecimal cpdm = obtenerCpdm(dom.getFechaFactura());  
 BigDecimal cpdam = obtenerCpdam(f, cpdm, t1, t0);  
 BigDecimal cidm = obtenerCidm(dom.getFechaFactura());  
 BigDecimal cidam = obtenerCidam(cpdam, t1, t0, cpdm, cidm);  
 cfcam = obtenerCFCam(cprciom, cpdam, cidam);  
 }  
  
 }  
 BigDecimal kc = obtenerKc(); //Capacidad demostrada  
 BigDecimal hm =  
 obtenerHm(dom.getFechaFactura().getMonth() + 1,  
 dom.getFechaFactura().getYear()); //Numero de horas en el mes  
 List<BigDecimal> xm =  
 obtenerXm(); //Lista de numero de horas por evento en el mes m en las cuales hubo evento que afecte la disponibilidad de la capacidad demostrada KC  
 BigDecimal ym = obtenerYm(hm, xm); //Numero de horas en el mes m en las cuales no hubo evento  
 List<BigDecimal> d = obtenerD(); //Lista de capacidades declaradas disponibles por evento  
 BigDecimal pdgm = obtenerPdgm(dom.getFechaFactura());  
 BigDecimal vmin = obtenerVmin(pdgm);  
 BigDecimal fcor = obtenerFcor(pdgm);  
 //List<BigDecimal> fdedmL = obtenerFdedmL(dom.getFechaFactura());  
  
 YearMonth mes = YearMonth.of(dom.getFechaFactura().getYear(), dom.getFechaFactura().getMonth());  
 BigDecimal pdd = obtenerPdd(dom.getCatPlanta()  
 .getPlantaId()  
 .intValue() == 1 ? Plantas.Aguila : Plantas.Sol, mes);  
 BigDecimal fadd = obtenerFadd(vmin, pdd, pdgm, fcor); //Factor de Ajuste por Disponibilidad Demostrada  
  
 BigDecimal faddm =  
 obtenerFaddm(fadd, hm, ym, xm, d, kc); //Factor de Ajuste por Disponibilidad Demostrada en el mes  
  
  
 BigDecimal pcfcm = obtenerPCFCm(cfcam, kc, faddm); //Pago del cargo fijo de capacidad en el mes m  
  
  
 DominioCalculoCargoFijo[] resultadoCalculos = new DominioCalculoCargoFijo[5];  
 DominioCalculoCargoFijo a = new DominioCalculoCargoFijo();  
 String pcfcmStr = "PCFCm\n$" + pcfcm.doubleValue() + " USD";  
 a.setKey(pcfcmStr);  
 a.setKey(pcfcmStr);  
 a.setName(pcfcmStr);  
 a.setColor("greengrad");  
 resultadoCalculos[0] = a;  
  
 DominioCalculoCargoFijo b = new DominioCalculoCargoFijo();  
 String CFDmxFADDmxKCm = "CFDm \* FADDm \* KCm\n" + pcfcm.doubleValue();  
 //b.setKey("CFDm x FADDm x KCm\nUS" + pcfcm);  
 b.setKey(CFDmxFADDmxKCm);  
 b.setName(CFDmxFADDmxKCm);  
 b.setParent(pcfcmStr);  
 b.setColor("greengrad");  
 b.setDir("left");  
 resultadoCalculos[1] = b;  
  
 DominioCalculoCargoFijo c = new DominioCalculoCargoFijo();  
 String cfcmStr = "CFCm\n" + cfcm.doubleValue();  
 c.setKey(cfcmStr);  
 c.setName(cfcmStr);  
 c.setParent(CFDmxFADDmxKCm);  
 c.setColor("greengrad");  
 resultadoCalculos[2] = c;  
  
 DominioCalculoCargoFijo dCF = new DominioCalculoCargoFijo();  
 String faddmStr = "FADDm\n" + faddm.doubleValue();  
 dCF.setKey(faddmStr);  
 dCF.setName(faddmStr);  
 dCF.setParent(CFDmxFADDmxKCm);  
 dCF.setColor("greengrad");  
 resultadoCalculos[3] = dCF;  
  
 DominioCalculoCargoFijo e = new DominioCalculoCargoFijo();  
 String kcm = "KCm\n" + kc.doubleValue();  
 e.setKey(kcm);  
 e.setName(kcm);  
 e.setParent(CFDmxFADDmxKCm);  
 e.setColor("greengrad");  
 resultadoCalculos[4] = e;  
  
 dom.setResultadoCalculos(resultadoCalculos);  
  
 return dom;  
 }  
  
  
};

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\buscadordedatos\BuscadorDeDatos.java

package mx.mbi.dgc.business.facturacion.facturador.buscadordedatos;  
  
import mx.mbi.dgc.dao.negocio.facturacion.DAOFacturacion;  
import mx.mbi.dgc.global.Globales.Plantas;  
  
import java.math.BigDecimal;  
  
import java.time.YearMonth;  
  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.HashMap;  
  
  
public class BuscadorDeDatos {  
 /\* HERENCIA \*/  
 DAOFacturacion daoFacturacion = new DAOFacturacion(); //Un DAO... to rule them all!  
 static Plantas planta;  
 private HashMap<String,BigDecimal> variablesManuales;  
 public BuscadorDeDatos(HashMap<String, BigDecimal> variablesManuales)  
 {  
 this.variablesManuales=variablesManuales;  
 }  
  
 public void setVariablesManuales(HashMap<String, BigDecimal> variablesManuales) {  
 this.variablesManuales = variablesManuales;  
 }  
  
 public HashMap<String, BigDecimal> getVariablesManuales() {  
 return variablesManuales;  
 }  
 //Para evitar que la calculadora sepa sobre el dao  
 public BigDecimal getFDEDm(Plantas planta, YearMonth mes) {  
 System.out.println(daoFacturacion);  
 return daoFacturacion.getFDEDm(planta, mes);  
 }  
  
 //Para evitar que la calculadora sepa sobre el dao  
 public void setFDEDm(Plantas planta, YearMonth mes, BigDecimal FDEDm) {  
 daoFacturacion.setFDEDm(planta, mes, FDEDm);  
 }  
  
  
 public BigDecimal getKC() {  
  
 return new BigDecimal("495000");  
 } //KC es constante, 495kW  
  
 public BigDecimal getHorasDelMes(YearMonth mes) {  
 int nMes = mes.getMonthValue();  
 int anio = mes.getYear();  
 int diferenciaDeAnios = anio - 2014;  
 nMes += diferenciaDeAnios \* 12;  
 return daoFacturacion.getHorasDelMes(nMes);  
 }  
  
 public BigDecimal getINPPm(YearMonth mes){  
 //TODO: este método es un mock-up  
 System.out.println(getVariablesManuales());  
 if(getVariablesManuales().get("INPPm")!=null)  
 return getVariablesManuales().get("INPPm");  
 return new BigDecimal("121.72594174551");  
 }  
  
 public BigDecimal getINPPo(){  
 if(getVariablesManuales().get("INPPo")!=null)  
 return getVariablesManuales().get("INPPo");  
 BigDecimal INPPo;  
 if (planta==Plantas.Aguila){  
 INPPo = new BigDecimal("50.323444664667");  
 } else {  
 INPPo = new BigDecimal("66.606055321941");  
 }  
  
 return INPPo;  
 }  
  
 public BigDecimal getUSPPIm(YearMonth mes){  
 //TODO: este método es un mock-up  
 if(getVariablesManuales().get("USPPIm")!=null)  
 return getVariablesManuales().get("USPPIm");  
 return new BigDecimal("194.800");  
 }  
  
 public BigDecimal getUSPPIo(){  
 BigDecimal USPPIo;  
 if(getVariablesManuales().get("USPPIo")!=null)  
 return getVariablesManuales().get("USPPIo");  
 if (planta==Plantas.Aguila){  
 USPPIo = new BigDecimal("123.600");  
 } else {  
 USPPIo = new BigDecimal("138.900");  
 }  
 return USPPIo;  
 }  
  
 //Para evitar que la calculadora sepa sobre el dao  
 public ArrayList<BigDecimal> getCargas(Plantas planta, YearMonth mes) {  
 return daoFacturacion.getCargas(planta, mes);  
 }  
  
  
 public BigDecimal getTCp(Date fechaDePago) {  
 //TODO: este método es un mock-up  
 if(getVariablesManuales().get("TCp")!=null)  
 return getVariablesManuales().get("TCp");  
 return new BigDecimal("19.1730");  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\buscadordedatos\BuscadorDeDatosAOMXLT.java

package mx.mbi.dgc.business.facturacion.facturador.buscadordedatos;  
  
import java.math.BigDecimal;  
import java.math.RoundingMode;  
  
import java.time.YearMonth;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
  
import mx.mbi.dgc.global.Globales;  
  
public class BuscadorDeDatosAOMXLT extends BuscadorDeDatos {  
 private YearMonth mes;  
  
 public BuscadorDeDatosAOMXLT(YearMonth mes, Globales.Plantas planta, HashMap<String, BigDecimal> variablesManuales) {  
 super(variablesManuales);  
 BuscadorDeDatos.planta = planta;  
 this.mes = mes;  
 }  
  
 public BigDecimal getCFOMdm() {  
  
 if (getVariablesManuales().get("CFOMdm") != null)  
 return getVariablesManuales().get("CFOMdm");  
 return daoFacturacion.getCFOMdm(planta, mes);  
 }  
  
 public BigDecimal getArranquesDelMes(YearMonth mes, int tipoDeCombustible, int tipoDeArranque) {  
 return daoFacturacion.getArranquesDelMes(planta, mes, tipoDeCombustible, tipoDeArranque);  
 }  
  
 public BigDecimal getHorasDeDiesel(YearMonth mes) {  
 ArrayList<BigDecimal> combustibles = daoFacturacion.getCombustibles(planta, mes);  
 BigDecimal horasDeDiesel = BigDecimal.ZERO;  
 BigDecimal fraccionDeHoraDeDiesel = BigDecimal.ZERO;  
 int cincominutalDeHora = 1;  
 for (BigDecimal combustible : combustibles) {  
 if (combustible.equals(new BigDecimal("2"))) {  
 fraccionDeHoraDeDiesel = fraccionDeHoraDeDiesel.add(BigDecimal.ONE);  
 }  
 if (cincominutalDeHora == 12) {  
 fraccionDeHoraDeDiesel = fraccionDeHoraDeDiesel.divide(new BigDecimal("12"), RoundingMode.HALF\_DOWN);  
 horasDeDiesel = horasDeDiesel.add(fraccionDeHoraDeDiesel);  
 cincominutalDeHora = 1;  
 continue;  
 }  
 cincominutalDeHora++;  
 }  
  
 return horasDeDiesel;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\buscadordedatos\BuscadorDeDatosPCA.java

package mx.mbi.dgc.business.facturacion.facturador.buscadordedatos;  
  
//import mx.mbi.dgc.dao.negocio.facturacion.DAOFacturacionPCA;  
import java.math.BigDecimal;  
  
import java.time.YearMonth;  
  
import java.util.HashMap;  
  
import mx.mbi.dgc.dao.negocio.facturacion.DAOFacturacionPCC;  
import mx.mbi.dgc.global.Globales;  
import mx.mbi.dgc.global.Globales.Plantas;  
  
/\*\*  
 \* buscador de datos para el cálculo del Pago por Cargo de Combustible  
 \*/  
public class BuscadorDeDatosPCA extends BuscadorDeDatos {  
  
 public BuscadorDeDatosPCA(HashMap<String,BigDecimal> variablesManuales) {  
 super(variablesManuales);  
 }  
 private DAOFacturacionPCC daoFacturacion = new DAOFacturacionPCC();  
  
 public BigDecimal getNAgmx(int x) {  
 //TODO: IMPLEMENTAR numero de arranques exitosos  
 if(getVariablesManuales().get("NAgmx")!=null)  
 return getVariablesManuales().get("NAgmx");  
 return new BigDecimal("2");  
 }  
  
 public BigDecimal getPPGm(Plantas planta, YearMonth mes) {  
 //TODO: maqueteado, falta implementación real  
 if(getVariablesManuales().get("PPGm")!=null)  
 return getVariablesManuales().get("PPGm");  
 return daoFacturacion.getPPGm(planta, mes);  
 }  
  
 public BigDecimal getCTRAgx(int x) {  
 if(getVariablesManuales().get("CTRAgx")!=null)  
 return getVariablesManuales().get("CTRAgx");  
 return getCTRA(x);  
  
 }  
  
 public BigDecimal getPCSm(Globales.Plantas planta, YearMonth mesAFacturar) {  
 //TODO: IMPLEMENTAR  
 if(getVariablesManuales().get("PCSm")!=null)  
 return getVariablesManuales().get("PCSm");  
 return new BigDecimal(2.221);  
 }  
  
 public BigDecimal getPCIm(Globales.Plantas planta, YearMonth mesAFacturar) {  
 //TODO: IMPLEMENTAR  
 if(getVariablesManuales().get("PCIm")!=null)  
 return getVariablesManuales().get("PCIm");  
 return new BigDecimal(2.221);  
 }  
  
 public BigDecimal getNAdmx(Globales.Plantas planta, YearMonth mesAFacturar, int x) {  
  
 //TODO: IMPLEMENTAR numero de arranques exitosos  
 if(getVariablesManuales().get("NAdmx")!=null)  
 return getVariablesManuales().get("NAdmx");  
 return new BigDecimal("2");  
 }  
  
 public BigDecimal getCTRAdx(Globales.Plantas planta, YearMonth mesAFacturar, int x) {  
 return getCTRA(x);  
  
 }  
  
 public BigDecimal getPVDm(Globales.Plantas planta, YearMonth mesAFacturar) {  
 //TODO: IMPLEMENTAR  
 if(getVariablesManuales().get("PVDm")!=null)  
 return getVariablesManuales().get("PVDm");  
 return new BigDecimal(2.221);  
 }  
  
 public BigDecimal getPCSdm(Globales.Plantas planta, YearMonth mesAFacturar) {  
 //TODO: IMPLEMENTAR  
 if(getVariablesManuales().get("PCSdm")!=null)  
 return getVariablesManuales().get("PCSdm");  
 return new BigDecimal(2.221);  
 }  
  
 public BigDecimal getPCIdm(Globales.Plantas planta, YearMonth mesAFacturar) {  
 //TODO: IMPLEMENTAR  
 if(getVariablesManuales().get("PCIdm")!=null)  
 return getVariablesManuales().get("PCIdm");  
 return new BigDecimal(2.221);  
 }  
  
 public BigDecimal getNAgmx(Globales.Plantas planta, YearMonth mesAFacturar, int x) {  
  
 //TODO: IMPLEMENTAR numero de arranques exitosos  
 if(getVariablesManuales().get("NAgmx")!=null)  
 return getVariablesManuales().get("NAgmx");  
 return new BigDecimal("2");  
 }  
  
 public BigDecimal getCTRAgx(Globales.Plantas planta, YearMonth mesAFacturar, int x) {  
 return getCTRA(x);  
 }  
  
 public BigDecimal getNAgmdx(Globales.Plantas planta, YearMonth mesAFacturar, int dia, int x) {  
 //TODO: IMPLEMENTAR numero de arranques exitosos  
 return new BigDecimal("2");  
 }  
  
 private BigDecimal getCTRA(int x) {  
  
 switch (x) {  
 case 1:  
 return new BigDecimal(4110000000f);  
 case 2:  
 return new BigDecimal(3507000000f);  
 case 3:  
 return new BigDecimal(580000000f);  
 default:  
 return null;  
 }  
 }  
  
 public BigDecimal getPPGmd(Globales.Plantas planta, YearMonth mesAFacturar, int dia) {  
 //TODO: IMPLEMENTAR  
 return new BigDecimal(2.221);  
 }  
  
 public BigDecimal getPCSmd(Globales.Plantas planta, YearMonth mesAFacturar, int dia) {  
 //TODO: IMPLEMENTAR  
 return new BigDecimal(2.221);  
 }  
  
 public BigDecimal getPCImd(Globales.Plantas planta, YearMonth mesAFacturar, int dia) {  
 //TODO: IMPLEMENTAR  
 return new BigDecimal(2.221);  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\buscadordedatos\BuscadorDeDatosPCC.java

package mx.mbi.dgc.business.facturacion.facturador.buscadordedatos;  
  
import java.math.BigDecimal;  
import java.math.RoundingMode;  
  
import java.time.YearMonth;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
  
import mx.mbi.dgc.dao.negocio.facturacion.DAOFacturacionPCC;  
import mx.mbi.dgc.global.Globales.Plantas;  
  
/\*\*  
 \* buscador de datos para el cálculo del Pago por Cargo de Combustible  
 \*/  
public class BuscadorDeDatosPCC extends BuscadorDeDatos {  
   
 public BuscadorDeDatosPCC(HashMap<String,BigDecimal> variablesManuales) {  
 super(variablesManuales);  
 }  
  
 private DAOFacturacionPCC daoFacturacion = new DAOFacturacionPCC();  
  
 /\*Para minimizar las llamadas a la Base de Datos,  
 guarda todos los valores del mes y luego devuelve sólo los que se necesiten por hora\*/  
 private ArrayList<BigDecimal> PCIs = null;  
 private ArrayList<BigDecimal> PCSs = null;  
 private ArrayList<BigDecimal> CTPs = null;  
 private ArrayList<BigDecimal> presiones = null;  
 private ArrayList<BigDecimal> temperaturas = null;  
 private ArrayList<BigDecimal> horasDeGas = null;  
 private ArrayList<BigDecimal> energias = null;  
 private ArrayList<BigDecimal> combustibles = null;  
  
   
 public BigDecimal getCAm(Plantas planta, YearMonth mes) {  
 //TODO: maqueteado, falta implementación real  
 return daoFacturacion.getCAm(planta, mes);  
 }  
  
 public BigDecimal getPPGm(Plantas planta, YearMonth mes) {  
 //TODO: maqueteado, falta implementación real  
 return daoFacturacion.getPPGm(planta, mes);  
 }  
  
 /\*\*  
 \* Devuelve el PCImh de la hora "h" del mes "m"  
 \*  
 \* @param planta planta para la cual se realiza la búsqueda  
 \* @param mes mes del cual se busca el PCImh  
 \* @param hora hora del mes  
 \* @return el valor de PCImh para la hora "h" del mes "m"  
 \*/  
 public BigDecimal getPCImh(Plantas planta, YearMonth mes, int hora) {  
 //Si no existe la lista, la recupera  
 if (PCIs == null) {  
 PCIs = daoFacturacion.getPCIm(planta, mes);  
 }  
  
 return calcularPromedioHorario(PCIs, hora);  
 }  
  
 public BigDecimal getPCIm(Plantas planta, YearMonth mes) {  
 //TODO: verificar si es correcto, de momento hace el promedio mensual  
 BigDecimal PCIm;  
 PCIs = daoFacturacion.getPCIm(planta, mes);  
  
 BigDecimal suma = new BigDecimal("0");  
 for (BigDecimal PCI : PCIs) {  
 suma = suma.add(PCI);  
 }  
  
 PCIm = suma.divide(new BigDecimal(PCIs.size()));  
  
 return PCIm;  
 }  
  
 /\*\*  
 \* Devuelve el PCSmh de la hora "h" del mes "m"  
 \*  
 \* @param planta planta para la cual se realiza la búsqueda  
 \* @param mes mes del cual se busca el PCImh  
 \* @param hora hora del mes  
 \* @return el valor de PCImh para la hora "h" del mes "m"  
 \*/  
 public BigDecimal getPCSmh(Plantas planta, YearMonth mes, int hora) {  
 //Si no existe la lista, la recupera  
 if (PCSs == null) {  
 PCSs = daoFacturacion.getPCSm(planta, mes);  
 }  
  
 return calcularPromedioHorario(PCSs, hora);  
 }  
  
 public BigDecimal getPCSm(Plantas planta, YearMonth mes) {  
 //TODO: verificar si es correcto, de momento hace el promedio mensual  
 BigDecimal PCSm;  
 PCSs = daoFacturacion.getPCSm(planta, mes);  
  
 BigDecimal suma = new BigDecimal("0");  
 for (BigDecimal PCS : PCSs) {  
 suma = suma.add(PCS);  
 }  
  
 PCSm = suma.divide(new BigDecimal(PCSs.size()));  
  
 return PCSm;  
 }  
  
  
 /\*\*  
 \* Numero de Horas Operando en Vacío con gas  
 \* @param planta planta para la cual se busca el NHOVg  
 \* @param mes mes que se factura  
 \* @param hora hora para la que se busca el NHOVg  
 \* @return la fracción de la hora en que la planta funcionó con gas  
 \*/  
 public BigDecimal getNHOVgmh(Plantas planta, YearMonth mes, int hora) {  
 if (combustibles == null) {  
 combustibles = daoFacturacion.getCombustibles(planta, mes);  
 }  
  
 ArrayList<BigDecimal> combustiblesPorHora = getValoresPorHora(combustibles, hora);  
 ArrayList<BigDecimal> cincominutalesGas = new ArrayList<>();  
  
 for (BigDecimal combustible : combustiblesPorHora) {  
 if (combustible.equals(new BigDecimal("1"))) {  
 cincominutalesGas.add(combustible);  
 }  
 }  
  
 return new BigDecimal(cincominutalesGas.size()).divide(new BigDecimal(12), RoundingMode.HALF\_DOWN);  
 }  
  
 public BigDecimal getNHOVdm(Plantas planta, YearMonth mes) {  
 if (combustibles == null) {  
 combustibles = daoFacturacion.getCombustibles(planta, mes);  
 }  
  
 ArrayList<BigDecimal> cincominutalesDiesel = new ArrayList<>();  
  
 for (BigDecimal combustible : combustibles) {  
 if (!combustible.equals(new BigDecimal("1"))) {  
 cincominutalesDiesel.add(combustible);  
 }  
 }  
  
 //Se divide entre 12 porque hay 12 horas por cincominutal.  
 return new BigDecimal(cincominutalesDiesel.size()).divide(new BigDecimal(12), RoundingMode.HALF\_DOWN);  
 }  
  
 public BigDecimal getCTRghCSh(Plantas planta, YearMonth mes, int hora) {  
 //TODO: IMPLEMENTAR - No hay valores de ejemplo ni encuentro a qué corresponde  
 /\*Definido como: CTUNG quemando Combustible Base (kJ/kWh), asociado a la Carga {Cdh}, de acuerdo a lo establecido  
 en la Propuesta en la Hora "h" del Día "d", ponderado de acuerdo al nivel de Carga obtenido mediante el Modelo  
 Matemático\*/  
  
 if(getVariablesManuales().get("CTRghCSh")!=null)  
 return getVariablesManuales().get("CTRghCSh");  
 return new BigDecimal(1);  
 }  
  
 public ArrayList<BigDecimal> getCTRdhCSh(Plantas planta, YearMonth mes) {  
 //TODO: IMPLEMENTAR - No hay valores de ejemplo ni encuentro a qué corresponde.  
 //TODO: DE momento se devuelve como una lista de "1".  
 /\*Definido como: CTUNG quemando Diesel, asociado a la Carga {CSh}, de acuerdo a lo establecido  
 en la Propuesta en la Hora "h" del mes, ponderado de acuerdo al nivel de Carga obtenido mediante el Modelo  
 Matemático\*/  
  
 ArrayList<BigDecimal> CTRs = new ArrayList<>();  
  
 int horasDelMes = getHorasDelMes(mes).intValue();  
  
 for (int hora = 0; hora < horasDelMes; hora++) {  
 CTRs.add(BigDecimal.ONE);  
 }  
  
 return CTRs;  
 }  
  
 /\*\*  
 \* Devuelve el promedio de Energías Activas de los cincominutales en los que se haya quemado gas dentro de una hora.  
 \* @param planta planta para la cual se va a hacer el cálculo  
 \* @param mes mes de facturación  
 \* @param hora hora de la cual se busca el Eg  
 \* @return promedio de Energías Activas de los cincominutales en los que se quemó gas de la hora dada.  
 \*/  
 public BigDecimal getEg(Plantas planta, YearMonth mes, int hora) {  
 //TODO: De momento calcula el promedio de la Energía Activa de los cincominutales en los que se quemó gas dentro de una hora dada  
 //TODO: REVISAR si es el dato que se debe usar  
 cargarEnergiasYCombustibles(planta, mes);  
  
 ArrayList<BigDecimal> energiasDeLaHora = getValoresPorHora(energias, hora);  
 ArrayList<BigDecimal> combustiblesDeLaHora = getValoresPorHora(combustibles, hora);  
  
 //Se recorre hacia atrás para evitar errores de índice al quitar elementos del ArrayList  
 for (int cincominutal = 11; cincominutal >= 0; cincominutal--) {  
 //Si el combustible no es gas (1), no se usa el valor  
 if (!combustiblesDeLaHora.get(cincominutal).equals(new BigDecimal("1"))) {  
 energiasDeLaHora.remove(cincominutal);  
 }  
 }  
  
 BigDecimal Eg = new BigDecimal("0");  
  
 for (BigDecimal energia : energiasDeLaHora) {  
 Eg = Eg.add(energia);  
 }  
  
 //TODO: Revisar si se busca el promedio horario de energías por cincominutal  
 Eg = Eg.divide(new BigDecimal(energiasDeLaHora.size()), RoundingMode.HALF\_DOWN);  
  
 return Eg;  
 }  
  
 /\*\*  
 \* Devuelve los valores por hora del mes de las Energías Activas quemando combustibles alternos.  
 \* @param planta planta para la que se genera la factura  
 \* @param mes mes a facturar  
 \* @return valores de Edh del mes, enlistados por hora  
 \*/  
 public ArrayList<BigDecimal> getEdh(Plantas planta, YearMonth mes) {  
 //TODO: De momento calcula el promedio de la Energía Activa de los cincominutales en los que se quemó gas, por hora  
 //TODO: REVISAR si es el dato que se debe usar  
 cargarEnergiasYCombustibles(planta, mes);  
  
 ArrayList<BigDecimal> Edhs = new ArrayList<>();  
 int horasDelMes = getHorasDelMes(mes).intValue();  
  
 System.out.printf("HORAS DEL MES: %d\n", horasDelMes);  
 System.out.printf("Energías encontradas: %d\n", energias.size());  
 System.out.printf("Combustibles encontrados: %d\n", combustibles.size());  
  
 for (int hora = 0; hora < horasDelMes; hora++) {  
 ArrayList<BigDecimal> energiasDeLaHora = getValoresPorHora(energias, hora);  
 ArrayList<BigDecimal> combustiblesDeLaHora = getValoresPorHora(combustibles, hora);  
  
 System.out.printf("Energias de la hora %d: %d\n" +  
 "Combustibles: %d\n",  
 hora, energiasDeLaHora.size(), combustiblesDeLaHora.size());  
  
 //Se recorre hacia atrás para evitar errores de índice al quitar elementos del ArrayList  
 for (int cincominutal = 11; cincominutal >= 0; cincominutal--) {  
 //Si el combustible no es gas (1), no se usa el valor  
 if (combustiblesDeLaHora.get(cincominutal).equals(new BigDecimal("1"))) {  
 energiasDeLaHora.remove(cincominutal);  
 }  
 }  
  
 BigDecimal Edh = new BigDecimal("0");  
  
 for (BigDecimal energia : energiasDeLaHora) {  
 Edh = Edh.add(energia);  
 }  
  
 if (energiasDeLaHora.size() != 0) {  
 Edhs.add(Edh.divide(new BigDecimal(energiasDeLaHora.size()), RoundingMode.HALF\_DOWN));  
 } else {  
 Edhs.add(BigDecimal.ZERO);  
 }  
 System.out.printf("Calculada la energía de la hora %d: %f en el registro %d\n", hora, Edhs.get(Edhs.size() - 1).floatValue(), Edhs.size());  
 }  
  
 System.out.printf("Se calcularon %s valores\n", Edhs.size());  
  
 return Edhs;  
 }  
  
 public BigDecimal getPVDm(YearMonth mes) {  
 //TODO: IMPLEMENTAR, es un mock-up  
 if(getVariablesManuales().get("PVDm")!=null)  
 return getVariablesManuales().get("PVDm");  
 return new BigDecimal("20540");  
 }  
  
 public ArrayList<BigDecimal> getPresionesPorHora(Plantas planta, int combustible, YearMonth mes, Integer hora) {  
  
 if (presiones == null||presiones.size()==0) {  
 //System.out.printf("Llamando al dao para leer presiones de la panta %d, combustible %d, mes %s, hora %d\n",  
 //planta.getID().intValue(), combustible, mes.toString(), hora);  
 presiones = daoFacturacion.leerCincominutalVarDelMes(planta, combustible, mes, 3, 10);  
 } else {  
 System.out.printf("Ya hay una lista de presiones cargadas con :%d elementos\n", presiones.size());  
 }  
  
 System.out.printf("Obteniendo presion barometrica de la hora %d\n" +  
 "Presiones totales del mes: %d\n", hora, presiones.size());  
  
 return getValoresPorHora(presiones, hora);  
 }  
  
 public ArrayList<BigDecimal> getPresiones(Plantas planta, int combustible, YearMonth mes) {  
 if (presiones == null||presiones.size()==0) {  
 presiones = daoFacturacion.leerCincominutalVarDelMes(planta, combustible, mes, 3, 10);  
 }  
  
 return presiones;  
 }  
  
 public ArrayList<BigDecimal> getTemperaturasPorHora(Plantas planta, int combustible, YearMonth mes, Integer hora) {  
  
 if (temperaturas == null||temperaturas.size()==0) {  
 temperaturas = daoFacturacion.leerCincominutalVarDelMes(planta, combustible, mes, 1, 14);  
 }  
  
 return getValoresPorHora(temperaturas, hora);  
 }  
  
 public ArrayList<BigDecimal> getTemperaturas(Plantas planta, int combustible, YearMonth mes) {  
  
 if (temperaturas == null||temperaturas.size()==0) {  
 temperaturas = daoFacturacion.leerCincominutalVarDelMes(planta, combustible, mes, 1, 14);  
 }  
  
 return temperaturas;  
 }  
  
 //Métodos privados  
  
 private ArrayList<BigDecimal> getValoresPorHora(ArrayList<BigDecimal> valoresDelMes, Integer hora) {  
 //TODO: Verificar que la hora es válida  
 System.out.printf("Extrayendo valores de la hora %d de %d valores del mes\n", hora, valoresDelMes.size());  
 ArrayList<BigDecimal> valoresDeLaHora = new ArrayList<>();  
 for (int cincomunutal = hora \* 12; cincomunutal < (hora + 1) \* 12; cincomunutal++) {  
 valoresDeLaHora.add(valoresDelMes.get(cincomunutal));  
 }  
  
 return valoresDeLaHora;  
 }  
  
 private BigDecimal calcularPromedioHorario(ArrayList<BigDecimal> listaDeValoresDelMes, int hora) {  
 //TODO: Verificar que la hora es válida  
 BigDecimal valor = new BigDecimal(0);  
  
 for (int cincominutal = hora \* 12; cincominutal < (hora + 1) \* 12; cincominutal++) {  
 if(listaDeValoresDelMes.get(cincominutal)!=null)  
 valor = valor.add(listaDeValoresDelMes.get(cincominutal));  
 }  
  
 valor = valor.divide(new BigDecimal(12), RoundingMode.HALF\_DOWN);  
  
 return valor;  
 }  
  
 private void cargarEnergiasYCombustibles(Plantas planta, YearMonth mes) {  
 if (energias == null) {  
 energias = daoFacturacion.getEnergias(planta, mes);  
 }  
 if (combustibles == null) {  
 combustibles = daoFacturacion.getCombustibles(planta, mes);  
 }  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\buscadordedatos\BuscadorDeDatosPCFC.java

package mx.mbi.dgc.business.facturacion.facturador.buscadordedatos;  
  
import java.math.BigDecimal;  
  
import java.time.YearMonth;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
  
import mx.mbi.dgc.business.facturacion.facturador.EventosDeDisponibilidad;  
import mx.mbi.dgc.global.Globales;  
import mx.mbi.dgc.global.Globales.Plantas;  
  
public class BuscadorDeDatosPCFC extends BuscadorDeDatos {  
  
 //private DAOFacturacion daoFacturacion = new DAOFacturacion();  
  
 private YearMonth mes;  
  
   
 public BuscadorDeDatosPCFC(YearMonth mes, Plantas planta, HashMap<String, BigDecimal> variablesManuales) {  
 super(variablesManuales);  
 this.mes = mes;  
 BuscadorDeDatos.planta = planta;  
 }  
  
 public ArrayList<EventosDeDisponibilidad> getEventosDeDisponibilidad() {  
 //TODO: implementar DAO, la versión actual del método es un mock-up  
 //return daoFacturacion.getEventosDeDisponibilidad(planta, mes);  
 return null;  
 }  
  
 public Object getCFCm(Globales.Plantas planta, YearMonth mesAFacturar) {  
 if(getVariablesManuales().get("CFCm")!=null)  
 return getVariablesManuales().get("CFCm");  
 return new BigDecimal("4.87235254498452");  
 }  
  
 public Object getT0() {  
 if(getVariablesManuales().get("T0")!=null)  
 return getVariablesManuales().get("T0");  
 return new BigDecimal("4.87235254498452");  
 }  
  
 public Object getT1() {  
 if(getVariablesManuales().get("T1")!=null)  
 return getVariablesManuales().get("T1");  
 return new BigDecimal("4.87235254498452");  
 }  
  
 public Object getPDGm(YearMonth mesAFacturar, Globales.Plantas planta) {  
 if(getVariablesManuales().get("PDGm")!=null)  
 return getVariablesManuales().get("PDGm");  
 return new BigDecimal("4.87235254498452");  
 }  
  
 public Object getCPRCIOm(Globales.Plantas planta, YearMonth mesAFacturar) {  
 if(getVariablesManuales().get("CPRCIOm")!=null)  
 return getVariablesManuales().get("CPRCIOm");  
 return new BigDecimal("4.87235254498452");  
 }  
  
 public Object getF\_CPDam(Globales.Plantas planta, YearMonth mesAFacturar) {  
 if(getVariablesManuales().get("FCPDam")!=null)  
 return getVariablesManuales().get("FCPDam");  
 return new BigDecimal("4.87235254498452");  
 }  
  
 public Object getCPDm(Globales.Plantas planta, YearMonth mesAFacturar) {  
 if(getVariablesManuales().get("CPDm")!=null)  
 return getVariablesManuales().get("CPDm");  
 return new BigDecimal("4.87235254498452");  
 }  
  
 public Object getCIDm(Globales.Plantas planta, YearMonth mesAFacturar) {  
 if(getVariablesManuales().get("CIDm")!=null)  
 return getVariablesManuales().get("CIDm");  
 return new BigDecimal("4.87235254498452");  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\buscadordedatos\BuscadorDeDatosPCFOM.java

package mx.mbi.dgc.business.facturacion.facturador.buscadordedatos;  
  
import java.math.BigDecimal;  
  
import java.time.YearMonth;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
  
import mx.mbi.dgc.business.facturacion.facturador.EventosDeDisponibilidad;  
import mx.mbi.dgc.global.Globales.Plantas;  
  
public class BuscadorDeDatosPCFOM extends BuscadorDeDatos{  
  
 //private DAOFacturacion daoFacturacion = new DAOFacturacion();  
  
 private YearMonth mes;  
  
  
 public BuscadorDeDatosPCFOM(YearMonth mes, Plantas planta, HashMap<String, BigDecimal> variablesManuales) {  
 super(variablesManuales);  
 setVariablesManuales(variablesManuales);  
 this.mes = mes;  
 BuscadorDeDatos.planta = planta;  
 }  
  
 public ArrayList<EventosDeDisponibilidad> getEventosDeDisponibilidad() {  
 //TODO: implementar DAO, la versión actual del método es un mock-up  
 //return daoFacturacion.getEventosDeDisponibilidad(planta, mes);  
 return null;  
 }  
  
 public BigDecimal getPDGm() {  
 if(getVariablesManuales().get("PDGm")!=null)  
 return getVariablesManuales().get("PDGm");  
 BigDecimal PDGm;  
 if (planta == Plantas.Aguila) {  
 PDGm = new BigDecimal(".9575");  
 } else {  
 //TODO: Revisar si es 93% o 0.93%  
 PDGm = new BigDecimal(".93");  
 }  
 return PDGm;  
 }  
  
 public BigDecimal getCFOMmtm() {  
 //TODO: este método es un mock-up  
  
 if(getVariablesManuales().get("CFOMmtm")!=null)  
 return getVariablesManuales().get("CFOMmtm");  
 BigDecimal CFOMmtm;  
 if (planta == Plantas.Aguila) {  
 CFOMmtm = new BigDecimal("1.9967");  
 } else {  
 CFOMmtm = new BigDecimal("3.347");  
 }  
 return CFOMmtm;  
 }  
  
 public BigDecimal getCFOMdm() {  
 //TODO: este método es un mock-up  
 if(getVariablesManuales().get("CFOMdm")!=null)  
 return getVariablesManuales().get("CFOMdm");  
 BigDecimal CFOMdm;  
 if (planta == Plantas.Aguila) {  
 CFOMdm = new BigDecimal("1.3400");  
 } else {  
 CFOMdm = new BigDecimal("1.6270");  
 }  
 return CFOMdm;  
 }  
  
 public BigDecimal getCFOMmom() {  
 //TODO: este método es un mock-up  
 if(getVariablesManuales().get("CFOMmom")!=null)  
 return getVariablesManuales().get("CFOMmom");  
 BigDecimal CFOMmom;  
 if (planta == Plantas.Aguila) {  
 CFOMmom = new BigDecimal("1.4734");  
 } else {  
 CFOMmom = new BigDecimal("4.175");  
 }  
 return CFOMmom;  
 }  
  
 public BigDecimal getIESm() {  
 //TODO: este método es un mock-up  
 if(getVariablesManuales().get("IESm")!=null)  
 return getVariablesManuales().get("IESm");  
 BigDecimal IESm;  
 if (planta == Plantas.Aguila) {  
 IESm = new BigDecimal("4.87235254498452");  
 } else {  
 IESm = new BigDecimal("2.13943035409537");  
 }  
 return IESm;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\buscadordedatos\BuscadorDeDatosPCVOM.java

package mx.mbi.dgc.business.facturacion.facturador.buscadordedatos;  
  
import java.math.BigDecimal;  
  
import java.time.YearMonth;  
  
import java.util.Date;  
import java.util.HashMap;  
  
import mx.mbi.dgc.global.Globales.Plantas;  
  
public class BuscadorDeDatosPCVOM extends BuscadorDeDatos {  
  
 //private DAOFacturacion daoFacturacion = new DAOFacturacion();  
  
 private YearMonth mes;  
  
 public BuscadorDeDatosPCVOM(YearMonth mes, Plantas planta, HashMap<String, BigDecimal> variablesManuales) {  
 super(variablesManuales);  
 this.mes = mes;  
 BuscadorDeDatos.planta = planta;  
 }  
  
  
 BigDecimal getINPPm() {  
 //TODO: este método es un mock-up  
 if(getVariablesManuales().get("INPPm")!=null)  
 return getVariablesManuales().get("INPPm");  
 return new BigDecimal("121.72594174551");  
 }  
  
  
 public BigDecimal getCVOMm(Date fecha) {  
 //TODO: este método es un mock-up  
 if(getVariablesManuales().get("CVOMm")!=null)  
 return getVariablesManuales().get("CVOMm");  
 return new BigDecimal("2.13943035409537");  
  
 }  
  
 public BigDecimal getPCAm(Date fecha) {  
 //TODO: este método es un mock-up  
 if(getVariablesManuales().get("PCAm")!=null)  
 return getVariablesManuales().get("PCAm");  
 return new BigDecimal("23.123");  
 }  
  
 public BigDecimal getCCAm(Date fecha) {  
 //TODO: este método es un mock-up  
 if(getVariablesManuales().get("CCAm")!=null)  
 return getVariablesManuales().get("CCAm");  
 return new BigDecimal("53.123");  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\calculadoradefacturas\CalculadoraAOMXLT.java

package mx.mbi.dgc.business.facturacion.facturador.calculadoradefacturas;  
  
import java.math.BigDecimal;  
import java.math.RoundingMode;  
  
import java.time.YearMonth;  
  
import java.util.Date;  
import java.util.HashMap;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosAOMXLT;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosAOMXLT.Variables;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.business.facturacion.facturador.buscadordedatos.BuscadorDeDatosAOMXLT;  
import mx.mbi.dgc.global.Globales.Plantas;  
  
public class CalculadoraAOMXLT extends CalculadoraDeFacturas{  
 private BuscadorDeDatosAOMXLT buscadorDeDatos;  
 private YearMonth mesAFacturar;  
 private Plantas planta;  
  
 @Override  
 public DatosDeFacturacion getDatosParaFacturacion(Plantas planta, YearMonth mesAFacturar, Date fechaDePago, HashMap<String,BigDecimal> variablesManuales) {  
 this.mesAFacturar = mesAFacturar;  
  
 buscadorDeDatos = new BuscadorDeDatosAOMXLT(mesAFacturar, planta,variablesManuales);  
  
 super.buscadorDeDatos.setVariablesManuales(variablesManuales);  
 datos = new DatosAOMXLT();  
  
 calcularVariablesIntermedias();  
 BigDecimal aomxlt=calcularAOMXLT();  
 guardarValorDeDato(Variables.AOMXLT,aomxlt);  
  
 guardarValorDeDato(DatosAOMXLT.Variables.monto,aomxlt);  
 BigDecimal montoFinal = (BigDecimal)leerValorDeDato(DatosAOMXLT.Variables.AOMXLT);  
 String montoEnLetras = obtenerCantidadEnLetras(montoFinal.multiply(new BigDecimal("0.16")));  
 guardarValorDeDato(Variables.MontoEnLetras, montoEnLetras);  
  
 return datos;  
 }  
  
 private BigDecimal calcularAOMXLT() {  
 BigDecimal CFOMdm = (BigDecimal) leerValorDeDato(Variables.CFOMdm);  
 BigDecimal KC = (BigDecimal) leerValorDeDato(Variables.KC);  
 BigDecimal USPPImo = (BigDecimal) leerValorDeDato(Variables.USPPImo);  
 BigDecimal TCp = (BigDecimal) leerValorDeDato(Variables.TCp);  
  
 BigDecimal AX = (BigDecimal) leerValorDeDato(Variables.AX);  
  
 System.out.printf("CFOMdm: %s", CFOMdm.toString());  
 BigDecimal AOMXLT = CFOMdm.multiply(new BigDecimal("12"))  
 .multiply(KC).multiply(USPPImo).multiply(TCp);  
  
 if (planta == Plantas.Sol) {  
 AOMXLT = AOMXLT.multiply(AX.multiply(new BigDecimal("0.002962")));  
 } else {  
 BigDecimal HX = (BigDecimal) leerValorDeDato(Variables.HX);  
 AOMXLT = AOMXLT.multiply(  
 HX.multiply(new BigDecimal("0.00005283"))  
 .add(AX.multiply(new BigDecimal("0.002962"))));  
 }  
  
 guardarValorDeDato(Variables.AOMXLT, AOMXLT);  
  
 return AOMXLT;  
  
 }  
  
 private void calcularVariablesIntermedias() {  
 BigDecimal CFOMdm = buscadorDeDatos.getCFOMdm();  
 BigDecimal KC = buscadorDeDatos.getKC();  
 BigDecimal USPPIm = buscadorDeDatos.getUSPPIm(mesAFacturar);  
 BigDecimal USPPIo = buscadorDeDatos.getUSPPIo();  
 BigDecimal USPPImo = USPPIm.divide(USPPIo, RoundingMode.HALF\_DOWN);  
 BigDecimal TCp = buscadorDeDatos.getTCp(new Date());  
 BigDecimal HX = calcularHX();  
 BigDecimal AX = calcularAX();  
  
 guardarValorDeDato(DatosAOMXLT.Variables.CFOMdm, CFOMdm);  
 guardarValorDeDato(DatosAOMXLT.Variables.KC, KC);  
 guardarValorDeDato(DatosAOMXLT.Variables.USPPIm, USPPIm);  
 guardarValorDeDato(DatosAOMXLT.Variables.USPPIo, USPPIo);  
 guardarValorDeDato(DatosAOMXLT.Variables.USPPImo, USPPImo);  
 guardarValorDeDato(DatosAOMXLT.Variables.TCp, TCp);  
 guardarValorDeDato(DatosAOMXLT.Variables.AX, AX);  
 // if (planta == Plantas.Aguila) {  
 guardarValorDeDato(DatosAOMXLT.Variables.HX, HX);  
 // }  
 }  
  
 private BigDecimal calcularAX() {  
 //TODO: Definir bien el rango a usar, se supone que es un año desde el aniversario  
 BigDecimal AX = BigDecimal.ZERO;  
 for (long i = 12; i > 0; i--) {  
 YearMonth mes = mesAFacturar.minusMonths(i);  
 BigDecimal NAgm1 = buscadorDeDatos.getArranquesDelMes(mes, 1, 1);  
 BigDecimal NAgm2 = buscadorDeDatos.getArranquesDelMes(mes, 1, 2);  
 BigDecimal NAgm3 = buscadorDeDatos.getArranquesDelMes(mes, 1, 3);  
 AX = AX.add(NAgm1).add(NAgm2).add(NAgm3);  
 if (planta == Plantas.Aguila) {  
 BigDecimal NAdm1 = buscadorDeDatos.getArranquesDelMes(mes, 2, 1);  
 BigDecimal NAdm2 = buscadorDeDatos.getArranquesDelMes(mes, 2, 2);  
 BigDecimal NAdm3 = buscadorDeDatos.getArranquesDelMes(mes, 2, 3);  
 AX = AX.add(NAdm1).add(NAdm2).add(NAdm3);  
 }  
 }  
  
 AX = AX.subtract(new BigDecimal("30"));  
  
 return AX;  
 }  
  
  
 private BigDecimal calcularHX(){  
 BigDecimal HX = BigDecimal.ZERO;  
  
 for (long i = 12; i > 0; i--) {  
 BigDecimal HDm = buscadorDeDatos.getHorasDeDiesel(mesAFacturar);  
 HX = HX.add(HDm);  
 }  
  
 HX = HX.subtract(new BigDecimal("360"));  
  
 return HX;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\calculadoradefacturas\CalculadoraDeFacturas.java

package mx.mbi.dgc.business.facturacion.facturador.calculadoradefacturas;  
  
import java.math.BigDecimal;  
import java.math.RoundingMode;  
  
import java.time.YearMonth;  
import java.time.temporal.ChronoUnit;  
  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.HashMap;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCFC.Variables;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCFOM;  
import mx.mbi.dgc.business.facturacion.facturador.buscadordedatos.BuscadorDeDatos;  
import mx.mbi.dgc.global.Globales.Plantas;  
import mx.mbi.dgc.utils.cantidadesaletras.CantidadesALetras;  
  
public abstract class CalculadoraDeFacturas {  
 /\* INTERFAZ \*/  
 public abstract DatosDeFacturacion getDatosParaFacturacion(Plantas planta, YearMonth mesAFacturar, Date fechaDePago,  
 HashMap<String, BigDecimal> variablesManuales);  
  
 public BuscadorDeDatos buscadorDeDatos = new BuscadorDeDatos(getVariablesManuales());  
 private HashMap<String,BigDecimal> variablesManuales;  
  
  
 void setVariablesManuales(HashMap<String, BigDecimal> variablesManuales) {  
 this.variablesManuales = variablesManuales;  
 }  
  
 HashMap<String, BigDecimal> getVariablesManuales() {  
 return variablesManuales;  
 }  
 //-- Propiedades  
 static DatosDeFacturacion datos;  
  
 //-- Métodos  
  
 /\*\*  
 \* Guarda un valor en los DatosDeFacturacion cuando es un valor único.  
 \* @param variable variable para la cual se almacenan datos  
 \* @param valor valor a almacenar  
 \*/  
 void guardarValorDeDato(DatosDeFacturacion.Variables variable, Object valor) {  
 if (valor instanceof BigDecimal) {  
 BigDecimal bdValor = (BigDecimal) valor;  
 System.out.printf("Guardando valor de la variable %s: %f...\n", variable.getNombre(), bdValor);  
 }  
  
 //Crea una ArrayList con el valor como único elemento y la inserta.  
 ArrayList<Object> valorAAgregar = new ArrayList<>();  
 valorAAgregar.add(valor);  
 datos.valores.put(variable, valorAAgregar);  
 }  
  
 /\*\*  
 \* Añade un valor en los DatosDeFacturación cuando es parte de una lista de valores guardada en una variable.  
 \* @param variable la variable para la cuál se guarda el valor en la lista  
 \* @param valor el valor a añadir  
 \*/  
 void guardarValorEnListaDeDatos(DatosDeFacturacion.Variables variable, Object valor) {  
  
 System.out.printf("Guardando valor en la lista de la variable %s: %f...\n", variable.getNombre(),  
 (BigDecimal) valor);  
 //Si no existe la lista, la crea.  
 datos.valores.computeIfAbsent(variable, k -> new ArrayList<>());  
  
 //Añade el valor a la lista.  
 datos.valores  
 .get(variable)  
 .add(valor);  
  
 }  
  
 /\*\*  
 \* Lee un dato de los DatosDeFacturacion cuando es un valor único.  
 \* @param variable variable en la cual está almacenado el dato  
 \* @return valor de la variable  
 \*/  
 Object leerValorDeDato(DatosDeFacturacion.Variables variable) {  
 System.out.printf("Leyendo el valor de la variable %s\n", variable.getNombre());  
 //Como NO es una lista, devuelve el valor de índice 0, que es el único  
 return datos.valores  
 .get(variable)  
 .get(0);  
 }  
  
 /\*\*  
 \* Calcula el Promedio de Disponibilidad Demostrada del mes "m"  
 \* @param planta la planta para la cual se hará e cáculo  
 \* @param mes el mes para el cual se hará el cálculo, "m"  
 \* @return el valor del PDDm  
 \*/  
 BigDecimal calcularPDDm(Plantas planta, YearMonth mes) {  
 System.out.printf("Calculando PDDm del mes %s\n", mes.toString());  
  
 //Calcula el PDDm como promedio de los últimos 12 FDEDm  
 ArrayList<BigDecimal> FDEDs = new ArrayList<>();  
  
  
 for (int i = 0; i < 12; i++) {  
 YearMonth mesARevisar = mes.minus(i, ChronoUnit.MONTHS);  
 BigDecimal FDED = calcularFDEDm(planta, mesARevisar);  
 guardarValorDeDato(Variables.FDEDm,FDED);  
  
 if (FDED != null) {  
 FDEDs.add(FDED);  
 } else {  
 System.out.printf("No existen datos de FDEDm para el mesAFacturar %s\n", mesARevisar.toString());  
 break;  
 }  
 }  
 BigDecimal promedioDeFDEDs = BigDecimal.ZERO;  
 BigDecimal cantidadDeValoresDeFDED =  
 BigDecimal.ZERO; //Lleva la cuenta en BigDecimal para poder dividir fácilmente.  
 for (BigDecimal FDED : FDEDs) {  
 promedioDeFDEDs = promedioDeFDEDs.add(FDED);  
 cantidadDeValoresDeFDED = cantidadDeValoresDeFDED.add(BigDecimal.ONE);  
 }  
 System.out.printf("Hay %d meses para el calculo del FDED\n", FDEDs.size());  
 if (cantidadDeValoresDeFDED.equals(BigDecimal.ZERO)) {  
 System.out.println("No hay FDEDs para el cálculo del PDDm");  
 return null;  
 }  
 promedioDeFDEDs = promedioDeFDEDs.divide(cantidadDeValoresDeFDED, RoundingMode.HALF\_DOWN);  
 return promedioDeFDEDs;  
 }  
  
 /\*\*  
 \* Si existe, retorna y si no, calcula el Factor de Disponibilidad Equivalente Demostrada para un mes "m".  
 \* Como el cálculo está indicado con datos por hora pero los datos se miden por cincominutal, funciona  
 \* haciendo primero los promedios horarios con los cincominutales existentes.  
 \* @param planta planta para la cual se realiza el cálculo  
 \* @param mes mes para el cual se realiza el cálculo.  
 \* @return valor del Factor de Disponibilidad Equivalente Demostrada para el mes "m"  
 \*/  
 BigDecimal calcularFDEDm(Plantas planta, YearMonth mes) {  
 //TODO: se supone que la carga es por hora pero la tenemos por cincominutal, hice los promedios por hora.  
 System.out.printf("Calculando FDEDm del mes %s\n", mes.toString());  
 //Verifica si ya fue calculado el FDEDm del mesAFacturar this.mesAFacturar, si sí, lo devuelve, si no, lo calcula y lo guarda en la BD.  
 BigDecimal FDEDm = buscadorDeDatos.getFDEDm(planta, mes);  
 if (FDEDm == null) {  
 BigDecimal bdHorasDelMes = buscadorDeDatos.getHorasDelMes(mes);  
 if (bdHorasDelMes != null) {  
 int horasDelMes = bdHorasDelMes.intValue();  
 System.out.printf("Horas del mes %s: %d\n", mes.toString(), horasDelMes);  
 FDEDm = BigDecimal.ONE.divide(bdHorasDelMes, 6,RoundingMode.HALF\_DOWN);  
 ArrayList<BigDecimal> cargasDelMes = buscadorDeDatos.getCargas(planta, mes);  
 if (horasDelMes \* 12 !=  
 cargasDelMes.size()) { //Debe haber 12 cincominutales por hora del mesAFacturar  
 System.out.printf("ERROR al calcular FDEDm: Hay %d cargas registradas para %d horas del mesAFacturar %s\n",  
 cargasDelMes.size(), horasDelMes, mes.toString());  
 return null;  
 } else {  
 System.out.printf("Hay %d cargas registradas para %d horas del mes a facturar %s\n",  
 cargasDelMes.size(), horasDelMes, mes.toString());  
 BigDecimal sumaDeCargas = BigDecimal.ZERO;  
 //Por cada hora del mesAFacturar hace un promedio de las cargas de sus cincominutales  
 for (int i = 0; i < horasDelMes; i++) {  
 BigDecimal promedioHorario = BigDecimal.ZERO;  
 for (int j = 0; j < 12; j++) {  
 promedioHorario = promedioHorario.add(cargasDelMes.get(12 \* i + j));  
 }  
 promedioHorario = promedioHorario.divide(new BigDecimal(12), RoundingMode.HALF\_DOWN);  
 promedioHorario = promedioHorario.divide(buscadorDeDatos.getKC(), RoundingMode.HALF\_DOWN);  
 sumaDeCargas = sumaDeCargas.add(promedioHorario);  
  
 }  
  
 System.out.println("FDEDm antes:"+FDEDm);  
 FDEDm = FDEDm.multiply(sumaDeCargas);  
 System.out.println("FDEDm:"+FDEDm);  
 System.out.println("sumaDeCargas:"+sumaDeCargas);  
 buscadorDeDatos.setFDEDm(planta, mes, FDEDm);  
 }  
 } else {  
 System.out.printf("No se encontraron las horas del mes para el mes %s\n", mes);  
 }  
 }  
 return FDEDm;  
 }  
  
 void calcularINPPmo(YearMonth mes) {  
 final int precisionINPP = 14;  
 BigDecimal INPPm = buscadorDeDatos.getINPPm(mes);  
 BigDecimal INPPo = buscadorDeDatos.getINPPo();  
  
 guardarValorDeDato(DatosPCFOM.Variables.INPPm, INPPm);  
 guardarValorDeDato(DatosPCFOM.Variables.INPPo, INPPo);  
  
  
 guardarValorDeDato(DatosPCFOM.Variables.INPP, INPPm.divide(INPPo, precisionINPP, RoundingMode.HALF\_DOWN));  
 }  
  
 /\*\*  
 \* Permite convertir un BigDecimal a letras.  
 \* @param cantidad valor a convertir  
 \* @return valor expresado en letras  
 \*/  
 String obtenerCantidadEnLetras(BigDecimal cantidad) {  
 return CantidadesALetras.convertirImporteALetras(cantidad.intValue());  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\calculadoradefacturas\CalculadoraINVALIDO.java

package mx.mbi.dgc.business.facturacion.facturador.calculadoradefacturas;  
  
import java.math.BigDecimal;  
  
import java.time.YearMonth;  
  
import java.util.Date;  
import java.util.HashMap;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.global.Globales;  
  
/\*\*  
 \* Cuando no se define un tipoDeFactura válido, no se genera ningún cálculo ni documento.  
 \*/  
public class CalculadoraINVALIDO extends CalculadoraDeFacturas {  
 public DatosDeFacturacion getDatosParaFacturacion(Globales.Plantas planta, YearMonth mesAFacturar, Date fechaDePago, HashMap<String, BigDecimal> variablesManuales){  
 System.out.println("Tipo de Factura Invalido");  
 return null;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\calculadoradefacturas\CalculadoraPCA.java

package mx.mbi.dgc.business.facturacion.facturador.calculadoradefacturas;  
  
import java.math.BigDecimal;  
  
import java.time.YearMonth;  
  
import java.util.Date;  
import java.util.HashMap;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosNulos;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCA;  
import mx.mbi.dgc.business.facturacion.facturador.buscadordedatos.BuscadorDeDatosPCA;  
import mx.mbi.dgc.global.Globales;  
  
public class CalculadoraPCA extends CalculadoraDeFacturas {  
  
 private CalculadoraPCAm calculadora;  
 static BuscadorDeDatosPCA buscadorDeDatos;  
 static YearMonth mesAFacturar;  
 static Globales.Plantas planta;  
  
 //Métodos package-private  
 public DatosDeFacturacion getDatosParaFacturacion(Globales.Plantas planta, YearMonth mesAFacturar, Date fechaDePago, HashMap<String, BigDecimal> variablesManuales) {  
 setVariablesManuales(variablesManuales);  
 this.mesAFacturar = mesAFacturar;  
 this.planta = planta;  
  
 buscadorDeDatos = new BuscadorDeDatosPCA(variablesManuales);  
 super.buscadorDeDatos.setVariablesManuales(variablesManuales);  
 datos = new DatosPCA();  
  
 switch (planta) {  
 case Invalida:  
 System.out.println("Planta inválida...");  
 return new DatosNulos();  
 case Aguila:  
 calculadora = new CalculadoraPCAmAguila();  
 break;  
 case Sol:  
 calculadora = new CalculadoraPCAmSol();  
 }  
  
 calculadora.calcularVariablesIntermedias();  
 BigDecimal PCAm= calculadora.calcularPCAm(planta);  
 guardarValorDeDato(DatosPCA.Variables.PCAm,PCAm );  
  
 guardarValorDeDato(DatosPCA.Variables.monto,PCAm);  
 return datos;  
 }  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\calculadoradefacturas\CalculadoraPCAm.java

package mx.mbi.dgc.business.facturacion.facturador.calculadoradefacturas;  
  
import mx.mbi.dgc.global.Globales;  
  
import java.math.BigDecimal;  
  
interface CalculadoraPCAm {  
 BigDecimal calcularPCAm(Globales.Plantas planta);  
 void calcularVariablesIntermedias();  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\calculadoradefacturas\CalculadoraPCAmAguila.java

package mx.mbi.dgc.business.facturacion.facturador.calculadoradefacturas;  
  
import java.math.BigDecimal;  
import java.math.RoundingMode;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCA.Variables;  
import mx.mbi.dgc.global.Globales;  
  
class CalculadoraPCAmAguila extends CalculadoraPCA implements CalculadoraPCAm {  
 @Override  
 public BigDecimal calcularPCAm(Globales.Plantas planta) {  
 System.out.println("Calculando PCAm para AGUILA...");  
  
 BigDecimal PCAm;  
 BigDecimal PCAGm = (BigDecimal) leerValorDeDato(Variables.PCAGm);  
 BigDecimal PCADm = (BigDecimal) leerValorDeDato(Variables.PCADm);  
  
 PCAm = PCAGm.add(PCADm);  
  
 return PCAm;  
 }  
  
 @Override  
 public void calcularVariablesIntermedias() {  
 guardarValorDeDato(Variables.PCAGm, calcularPCAGm());  
 guardarValorDeDato(Variables.PCADm, calcularPCADm());  
 }  
  
  
 private BigDecimal calcularPCAGm() {  
 guardarValorDeDato(Variables.CTPAgm, calcularCTPAgm());  
 guardarValorDeDato(Variables.PCSm, calcularPCSm());  
 guardarValorDeDato(Variables.PCIm, calcularPCIm());  
 guardarValorDeDato(Variables.PPGm, buscadorDeDatos.getPPGm(planta, mesAFacturar));  
  
 BigDecimal CTPAgm = (BigDecimal) leerValorDeDato(Variables.CTPAgm);  
 BigDecimal PCSm = (BigDecimal) leerValorDeDato(Variables.PCSm);  
 BigDecimal PCIm = (BigDecimal) leerValorDeDato(Variables.PCIm);  
 BigDecimal PPGm = (BigDecimal) leerValorDeDato(Variables.PPGm);  
  
 return CTPAgm.multiply(PPGm.multiply(PCSm.divide(PCIm, 15, RoundingMode.HALF\_UP)));  
 }  
  
 private BigDecimal calcularPCADm() {  
 guardarValorDeDato(Variables.CTPAdm, calcularCTPAdm());  
 guardarValorDeDato(Variables.PVDm, calcularPVDm());  
 guardarValorDeDato(Variables.PCSdm, calcularPCSdm());  
 guardarValorDeDato(Variables.PCIdm, calcularPCIdm());  
  
 BigDecimal CTPAdm = (BigDecimal) leerValorDeDato(Variables.CTPAgm);  
 BigDecimal PVDm = (BigDecimal) leerValorDeDato(Variables.PCSm);  
 BigDecimal PCIdm = (BigDecimal) leerValorDeDato(Variables.PCIm);  
 BigDecimal PCSdm = (BigDecimal) leerValorDeDato(Variables.PPGm);  
  
 return CTPAdm.multiply(PVDm.multiply(PCSdm.divide(PCIdm, 15, RoundingMode.HALF\_UP)));  
 }  
  
 private BigDecimal calcularCTPAgm() {  
 BigDecimal CTPAgm = new BigDecimal(0);  
 for (int x = 1; x < 4; x++) {  
 CTPAgm = CTPAgm.add(buscadorDeDatos.getNAgmx(x).multiply(buscadorDeDatos.getCTRAgx(x)));  
 }  
  
 return CTPAgm;  
 }  
  
 private BigDecimal calcularPCSm() {  
 return (BigDecimal) buscadorDeDatos.getPCSm(planta, mesAFacturar);  
  
 }  
  
 private BigDecimal calcularPCIm() {  
 return buscadorDeDatos.getPCIm(planta, mesAFacturar);  
 }  
  
 private BigDecimal calcularCTPAdm() {  
 BigDecimal CTPAdm = new BigDecimal(0);  
 for (int x = 1; x < 4; x++) {  
 CTPAdm =  
 CTPAdm.add(buscadorDeDatos.getNAdmx(planta, mesAFacturar, x)  
 .multiply(buscadorDeDatos.getCTRAdx(planta, mesAFacturar, x)));  
 }  
  
 return CTPAdm;  
 }  
  
 private BigDecimal calcularPVDm() {  
  
 return buscadorDeDatos.getPVDm(planta, mesAFacturar);  
 }  
  
 private BigDecimal calcularPCSdm() {  
 return buscadorDeDatos.getPCSdm(planta, mesAFacturar);  
 }  
  
 private BigDecimal calcularPCIdm() {  
 return buscadorDeDatos.getPCIdm(planta, mesAFacturar);  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\calculadoradefacturas\CalculadoraPCAmSol.java

package mx.mbi.dgc.business.facturacion.facturador.calculadoradefacturas;  
  
import java.math.BigDecimal;  
import java.math.RoundingMode;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCA.Variables;  
import mx.mbi.dgc.global.Globales;  
import mx.mbi.dgc.utils.DateUtil;  
  
class CalculadoraPCAmSol extends CalculadoraPCA implements CalculadoraPCAm {  
 @Override  
 public BigDecimal calcularPCAm(Globales.Plantas planta) {  
 System.out.println("Calculando PCAm para SOL...");  
 BigDecimal PCAm = new BigDecimal(0);  
 for (int i = 1; i <= DateUtil.diasDelMes(mesAFacturar.getMonth().getValue(), mesAFacturar.getYear()); i++) {  
 PCAm = PCAm.add(calcularPCAGmd(i));  
 }  
  
 return PCAm;  
 }  
  
 @Override  
 public void calcularVariablesIntermedias() {  
 guardarValorDeDato(Variables.PCAGm, calcularPCAGm());  
 guardarValorDeDato(Variables.PCADm, calcularPCADm());  
 }  
  
  
 private BigDecimal calcularPCAGm() {  
 guardarValorDeDato(Variables.CTPAgm, calcularCTPAgm());  
 guardarValorDeDato(Variables.PCSm, calcularPCSm());  
 guardarValorDeDato(Variables.PCIm, calcularPCIm());  
 guardarValorDeDato(Variables.PPGm, buscadorDeDatos.getPPGm(planta, mesAFacturar));  
  
 BigDecimal CTPAgm = (BigDecimal) leerValorDeDato(Variables.CTPAgm);  
 BigDecimal PCSm = (BigDecimal) leerValorDeDato(Variables.PCSm);  
 BigDecimal PCIm = (BigDecimal) leerValorDeDato(Variables.PCIm);  
 BigDecimal PPGm = (BigDecimal) leerValorDeDato(Variables.PPGm);  
  
 return CTPAgm.multiply(PPGm.multiply(PCSm.divide(PCIm, 15, RoundingMode.HALF\_UP)));  
 }  
  
 private BigDecimal calcularPCADm() {  
 guardarValorDeDato(Variables.CTPAdm, calcularCTPAdm());  
 guardarValorDeDato(Variables.PVDm, calcularPVDm());  
 guardarValorDeDato(Variables.PCSdm, calcularPCSdm());  
 guardarValorDeDato(Variables.PCIdm, calcularPCIdm());  
  
 BigDecimal CTPAdm = (BigDecimal) leerValorDeDato(Variables.CTPAgm);  
 BigDecimal PVDm = (BigDecimal) leerValorDeDato(Variables.PCSm);  
 BigDecimal PCIdm = (BigDecimal) leerValorDeDato(Variables.PCIm);  
 BigDecimal PCSdm = (BigDecimal) leerValorDeDato(Variables.PPGm);  
  
 return CTPAdm.multiply(PVDm.multiply(PCSdm.divide(PCIdm, 15, RoundingMode.HALF\_UP)));  
 }  
  
 private BigDecimal calcularCTPAgm() {  
 BigDecimal CTPAgm = new BigDecimal(0);  
 for (int x = 1; x < 4; x++) {  
 CTPAgm =  
 CTPAgm.add(buscadorDeDatos.getNAgmx(planta, mesAFacturar, x)  
 .multiply(buscadorDeDatos.getCTRAgx(planta, mesAFacturar, x)));  
 }  
  
 return CTPAgm;  
 }  
  
 private BigDecimal calcularCTPAgmd(int dia) {  
 BigDecimal CTPAgmd = new BigDecimal(0);  
 for (int x = 1; x < 4; x++) {  
 CTPAgmd =  
 CTPAgmd.add(buscadorDeDatos.getNAgmdx(planta, mesAFacturar, dia, x)  
 .multiply(buscadorDeDatos.getCTRAgx(planta, mesAFacturar, x)));  
 }  
  
 return CTPAgmd;  
 }  
  
 private BigDecimal calcularPCSm() {  
 return buscadorDeDatos.getPCSm(planta, mesAFacturar);  
  
 }  
  
 private BigDecimal calcularPCIm() {  
 return buscadorDeDatos.getPCIm(planta, mesAFacturar);  
 }  
  
 private BigDecimal calcularCTPAdm() {  
 BigDecimal CTPAdm = new BigDecimal(0);  
 for (int x = 1; x < 4; x++) {  
 CTPAdm =  
 CTPAdm.add(buscadorDeDatos.getNAdmx(planta, mesAFacturar, x)  
 .multiply(buscadorDeDatos.getCTRAdx(planta, mesAFacturar, x)));  
 }  
  
 return CTPAdm;  
 }  
  
 private BigDecimal calcularPVDm() {  
  
 return buscadorDeDatos.getPVDm(planta, mesAFacturar);  
 }  
  
 private BigDecimal calcularPCSdm() {  
 return buscadorDeDatos.getPCSdm(planta, mesAFacturar);  
 }  
  
 private BigDecimal calcularPCIdm() {  
 return buscadorDeDatos.getPCSdm(planta, mesAFacturar);  
 }  
  
 private BigDecimal calcularPCAGmd(int dia) {  
 BigDecimal CTPAgmd = calcularCTPAgmd(dia);  
 BigDecimal PPGmd = calcularPPGmd(dia);  
 BigDecimal PCSmd = calcularPCSmd(dia);  
 BigDecimal PCImd = calcularPCImd(dia);  
 return CTPAgmd.multiply(PPGmd.multiply(PCSmd.divide(PCImd, 15, RoundingMode.HALF\_UP)));  
 }  
  
 private BigDecimal calcularPPGmd(int dia) {  
 return buscadorDeDatos.getPPGmd(planta, mesAFacturar, dia);  
  
 }  
  
 private BigDecimal calcularPCSmd(int dia) {  
  
 return buscadorDeDatos.getPCSmd(planta, mesAFacturar, dia);  
 }  
  
 private BigDecimal calcularPCImd(int dia) {  
 return buscadorDeDatos.getPCImd(planta, mesAFacturar, dia);  
  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\calculadoradefacturas\CalculadoraPCC.java

package mx.mbi.dgc.business.facturacion.facturador.calculadoradefacturas;  
  
import java.math.BigDecimal;  
import java.math.RoundingMode;  
  
import java.time.YearMonth;  
  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.HashMap;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosNulos;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCC;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCC.Variables;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCFOM;  
import mx.mbi.dgc.business.facturacion.facturador.buscadordedatos.BuscadorDeDatosPCC;  
import mx.mbi.dgc.global.Globales;  
  
public class CalculadoraPCC extends CalculadoraDeFacturas {  
  
 private CalculadoraPCCm calculadora;  
 static BuscadorDeDatosPCC buscadorDeDatos;  
 static YearMonth mesAFacturar;  
 static Globales.Plantas planta;  
  
 //Métodos package-private  
 public DatosDeFacturacion getDatosParaFacturacion(Globales.Plantas planta, YearMonth mesAFacturar, Date fechaDePago, HashMap<String, BigDecimal> variablesManuales) {  
 setVariablesManuales(variablesManuales);  
 CalculadoraPCC.mesAFacturar = mesAFacturar;  
 CalculadoraPCC.planta = planta;  
  
 System.out.printf("Buscando datos para la planta %d, mes %s\n", planta.getID().intValue(), mesAFacturar.toString());  
 buscadorDeDatos = new BuscadorDeDatosPCC(variablesManuales);  
 super.buscadorDeDatos.setVariablesManuales(variablesManuales);  
 datos = new DatosPCC();  
  
 switch (planta) {  
 case Invalida:  
 System.out.println("Planta inválida...");  
 return new DatosNulos();  
 case Aguila:  
 calculadora = new CalculadoraPCCmAguila();  
 break;  
 case Sol:  
 calculadora = new CalculadoraPCCmSol();  
 }  
  
 System.out.print("Calculando variables intermedias\n");  
 calculadora.calcularVariablesIntermedias();  
  
  
 BigDecimal PCCm= calculadora.calcularPCCm(planta);  
 guardarValorDeDato(DatosPCC.Variables.PCCm,PCCm );  
  
 guardarValorDeDato(DatosPCC.Variables.monto,PCCm);  
  
 BigDecimal montoFinal = (BigDecimal)leerValorDeDato(Variables.PCCm);  
 String montoEnLetras = obtenerCantidadEnLetras(montoFinal.multiply(new BigDecimal("0.16")));  
 guardarValorDeDato(DatosPCFOM.Variables.MontoEnLetras, montoEnLetras);  
  
 return datos;  
 }  
  
 BigDecimal calcularPGm() {  
 BigDecimal PGm;  
 BigDecimal PPGm = calcularPPGm();  
 guardarValorDeDato(Variables.PPGm, PPGm);  
 int horasDelMes = buscadorDeDatos.getHorasDelMes(mesAFacturar).intValue();  
  
 BigDecimal suma = new BigDecimal(0);  
  
 /\*Suma de todos los CTPgmh multiplicados por el cociente de PCSmh y PCImh para la hora "h" del mes "m",  
 multiplicado por el PPGm del mes. Por lo tanto se pueden sumar los CTP, PCS y PCI y luego multiplicar  
 por PPGm.\*/  
 for (int hora = 1; hora < horasDelMes; hora++) {  
 //CTPmh, PCImh y PCSmh se guardan como lista de valores por hora  
 BigDecimal PCSmh = buscadorDeDatos.getPCSmh(planta, mesAFacturar, hora);  
 BigDecimal PCImh = buscadorDeDatos.getPCImh(planta, mesAFacturar, hora);  
 BigDecimal CTPmh = calcularCTPmh(hora);  
 guardarValorEnListaDeDatos(Variables.PCSmh, PCSmh);  
 guardarValorEnListaDeDatos(Variables.PCImh, PCImh);  
 guardarValorEnListaDeDatos(Variables.CTPgmh, CTPmh);  
 //Realiza el cálculo de la suma de los CTPgmh por el cociente de PCSmh y PCImh  
 BigDecimal cocienteDePoderCalorifico = PCImh.divide(PCSmh, RoundingMode.HALF\_DOWN);  
 suma = suma.add(cocienteDePoderCalorifico.multiply(CTPmh));  
 }  
  
 PGm = suma.multiply(PPGm);  
  
 return PGm;  
 }  
  
 BigDecimal calcularCAm() {  
 //TODO: maqueteado, falta implementación real  
 return (BigDecimal) buscadorDeDatos.getCAm(planta, mesAFacturar);  
 }  
  
 private BigDecimal calcularPPGm(){  
 //TODO: maqueteado, falta implementación real  
 return buscadorDeDatos.getPPGm(planta, mesAFacturar);  
 }  
  
 private BigDecimal calcularCTPmh(int nHora) {  
 BigDecimal CTPmh;  
 System.out.println("Buscando NHOVgmh");  
 BigDecimal NHOVgmh = buscadorDeDatos.getNHOVgmh(planta, mesAFacturar, nHora);  
 System.out.println("Calculando CTOVg");  
 BigDecimal CTOVg = calcularCTOVg(nHora);  
 BigDecimal CTRghCSh = buscadorDeDatos.getCTRghCSh(planta, mesAFacturar, nHora);  
 //TODO: Definir si se usa la Energía Activa, la Energía Aparente o la Carga del Cincominutal para Eg  
 BigDecimal Egh = buscadorDeDatos.getEg(planta, mesAFacturar, nHora);  
  
 guardarValorEnListaDeDatos(Variables.NHOVgmh, NHOVgmh);  
 guardarValorEnListaDeDatos(Variables.CTOVg, CTOVg);  
  
 guardarValorEnListaDeDatos(Variables.CTRghCSh,CTRghCSh);  
 guardarValorEnListaDeDatos(Variables.Egh, Egh);  
  
 CTPmh = NHOVgmh.multiply(CTOVg).add(CTRghCSh.multiply(Egh));  
  
 guardarValorEnListaDeDatos(Variables.CTPgmh, CTPmh);  
  
 return CTPmh;  
 }  
  
 private BigDecimal calcularCTOVg(int nHora) {  
 //TODO: Ajuste para que use sólo los valores de cuando la planta está quemando Gas  
 BigDecimal CTOVg = new BigDecimal(0);  
 ArrayList<BigDecimal> presionBarometrica = buscadorDeDatos.getPresionesPorHora(planta, 1, mesAFacturar, nHora);  
 ArrayList<BigDecimal> temperaturaDeBulboSeco = buscadorDeDatos.getTemperaturasPorHora(planta, 1, mesAFacturar, nHora);  
  
 for (int cincominutal = 1; cincominutal < 12; cincominutal++) {  
 BigDecimal divisor = new BigDecimal(12);  
 BigDecimal dividendo = new BigDecimal("162.3295").multiply(new BigDecimal(1000000));  
 divisor = divisor.multiply(new BigDecimal("0.00052215")  
 .add(new BigDecimal("0.98751147").multiply(presionBarometrica.get(cincominutal))));  
 divisor = divisor.multiply(new BigDecimal("1.15516025")  
 .add(new BigDecimal(-0.00611085).multiply(temperaturaDeBulboSeco.get(cincominutal))));  
  
 CTOVg = CTOVg.add(dividendo.divide(divisor, RoundingMode.HALF\_DOWN));  
 }  
  
 return CTOVg;  
 }  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\calculadoradefacturas\CalculadoraPCCm.java

package mx.mbi.dgc.business.facturacion.facturador.calculadoradefacturas;  
  
import mx.mbi.dgc.global.Globales;  
  
import java.math.BigDecimal;  
  
interface CalculadoraPCCm {  
 BigDecimal calcularPCCm(Globales.Plantas planta);  
 void calcularVariablesIntermedias();  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\calculadoradefacturas\CalculadoraPCCmAguila.java

package mx.mbi.dgc.business.facturacion.facturador.calculadoradefacturas;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCC.Variables;  
  
import mx.mbi.dgc.global.Globales;  
  
import java.math.BigDecimal;  
import java.math.RoundingMode;  
import java.util.ArrayList;  
  
class CalculadoraPCCmAguila extends CalculadoraPCC implements CalculadoraPCCm{  
 @Override  
 public BigDecimal calcularPCCm(Globales.Plantas planta) {  
 System.out.println("Calculando PCCm para AGUILA...");  
  
 BigDecimal PCCm;  
 BigDecimal PGm = (BigDecimal) leerValorDeDato(Variables.PGm);  
 BigDecimal CAm = (BigDecimal) leerValorDeDato(Variables.CAm);  
 BigDecimal PDm = (BigDecimal) leerValorDeDato(Variables.PDm);  
  
 PCCm = PGm.add(PDm).add(CAm);  
  
 return PCCm;  
 }  
  
 @Override  
 public void calcularVariablesIntermedias() {  
 System.out.println("Calculando PDm");  
 guardarValorDeDato(Variables.PDm, calcularPDm());  
 System.out.println("Calculando PGm");  
 guardarValorDeDato(Variables.PGm, calcularPGm());  
 System.out.println("Calculando CAm");  
 guardarValorDeDato(Variables.CAm, calcularCAm());  
 }  
  
 //-- Métodos privados  
 private BigDecimal calcularPDm() {  
 BigDecimal PDm;  
 BigDecimal PVDm = buscadorDeDatos.getPVDm(mesAFacturar);  
 BigDecimal PCIDm = buscadorDeDatos.getPCIm(planta, mesAFacturar);  
 BigDecimal PCSDm = buscadorDeDatos.getPCSm(planta, mesAFacturar);  
 BigDecimal CTPdm = calcularCTPdm();  
  
 guardarValorDeDato(Variables.PVDm, PVDm);  
 guardarValorDeDato(Variables.PCIDm, PCIDm);  
 guardarValorDeDato(Variables.PCSDm, PCSDm);  
 guardarValorDeDato(Variables.CTPdm, CTPdm);  
  
 if (CTPdm == null) {  
 System.out.println("No se puede calcular PDm porque no se pudo calcular CTPdm");  
 return null;  
 }  
  
 PDm = CTPdm.multiply(PVDm).multiply(PCSDm.divide(PCIDm, RoundingMode.HALF\_DOWN));  
  
 return PDm;  
 }  
  
 private BigDecimal calcularCTPdm() {  
 BigDecimal CTPdm;  
 BigDecimal NHOVdm = buscadorDeDatos.getNHOVdm(planta, mesAFacturar);  
 guardarValorDeDato(Variables.NHOVdm, NHOVdm);  
 BigDecimal CTOVd = calcularCTOVd();  
 ArrayList<BigDecimal> CTRdhCSh = buscadorDeDatos.getCTRdhCSh(planta, mesAFacturar);  
 ArrayList<BigDecimal> Edh = buscadorDeDatos.getEdh(planta, mesAFacturar);  
  
 if (CTRdhCSh.size() != Edh.size()) {  
 System.out.printf("La cantidad de valores de CTRdh{CSh} (%d) y de EDh (%d) son diferentes\n" +  
 "No se puede calcular el CTPdm.\n", CTRdhCSh.size(), Edh.size());  
  
 return null;  
 }  
  
 //Suma todos los CTRdhCSh y los Edh del mes  
 BigDecimal suma = BigDecimal.ZERO;  
 for (int hora = 0; hora < CTRdhCSh.size(); hora++) {  
 suma = suma.add(CTRdhCSh.get(hora).multiply(Edh.get(hora)));  
  
 }  
  
 if (CTOVd == null) {  
 System.out.println("No se puede calcular CTPdm porque no se pudo calcular CTOVd");  
 return null;  
 }  
  
 CTPdm = NHOVdm.multiply(CTOVd).add(suma);  
  
 return CTPdm;  
 }  
  
 private BigDecimal calcularCTOVd() {  
 // Como ya se multiplica por NHOVdm, simplemente calcula la suma de todos los CTOV de los cincominutales  
 // en los que se quemó combustible alterno.  
  
 BigDecimal dividendo = new BigDecimal("170.665").multiply(new BigDecimal("1000000"));  
  
 ArrayList<BigDecimal> presionesBarometricas = buscadorDeDatos.getPresiones(planta, 2, mesAFacturar);  
 ArrayList<BigDecimal> temperaturasDeBulboSeco = buscadorDeDatos.getTemperaturas(planta, 2, mesAFacturar);  
 int horasDelMes = buscadorDeDatos.getHorasDelMes(mesAFacturar).intValue();  
  
 if (presionesBarometricas.size() != temperaturasDeBulboSeco.size()) {  
 System.out.println("No existen la misma cantidad de valores para \n" +  
 "presión barométrica y temperatura de bulbo seco\n" +  
 "No se puede calcular CTOVd");  
 return null;  
 }  
  
 BigDecimal suma = BigDecimal.ZERO;  
  
 for (int i = 0; i < presionesBarometricas.size(); i++) {  
 BigDecimal divisor = new BigDecimal("12")  
 .multiply(new BigDecimal("0.98751147").multiply(presionesBarometricas.get(i)).add(new BigDecimal("0.00052215")))  
 .multiply(new BigDecimal("-0.00611085").multiply(temperaturasDeBulboSeco.get(i)).add(new BigDecimal("1.15587201")));  
 suma = suma.add(BigDecimal.ONE.divide(divisor, RoundingMode.HALF\_DOWN));  
 }  
  
 BigDecimal CTOVd = suma.multiply(dividendo);  
  
 return CTOVd;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\calculadoradefacturas\CalculadoraPCCmSol.java

package mx.mbi.dgc.business.facturacion.facturador.calculadoradefacturas;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCC;  
import mx.mbi.dgc.global.Globales;  
  
import java.math.BigDecimal;  
  
class CalculadoraPCCmSol extends CalculadoraPCC implements CalculadoraPCCm {  
 @Override  
 public BigDecimal calcularPCCm(Globales.Plantas planta) {  
 System.out.println("Calculando PCCm para SOL...");  
  
 BigDecimal PCCm;  
 BigDecimal PGm = (BigDecimal) leerValorDeDato(DatosPCC.Variables.PGm);  
 BigDecimal CAm = (BigDecimal) leerValorDeDato(DatosPCC.Variables.CAm);  
  
 PCCm = PGm.add(CAm);  
  
 return PCCm;  
 }  
  
 @Override  
 public void calcularVariablesIntermedias() {  
 guardarValorDeDato(DatosPCC.Variables.PGm, calcularPGm());  
 guardarValorDeDato(DatosPCC.Variables.CAm, calcularCAm());  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\calculadoradefacturas\CalculadoraPCFC.java

package mx.mbi.dgc.business.facturacion.facturador.calculadoradefacturas;  
  
import java.math.BigDecimal;  
import java.math.RoundingMode;  
  
import java.time.YearMonth;  
import java.time.ZoneId;  
  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.List;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCFC;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCFC.Variables;  
import mx.mbi.dgc.business.facturacion.facturador.EventosDeDisponibilidad;  
import mx.mbi.dgc.business.facturacion.facturador.buscadordedatos.BuscadorDeDatosPCFC;  
import mx.mbi.dgc.global.Globales;  
import mx.mbi.dgc.global.Globales.Plantas;  
import mx.mbi.dgc.jpa.entity.TxEvento;  
import mx.mbi.dgc.utils.DateUtil;  
  
import org.joda.time.DateTime;  
import org.joda.time.Period;  
  
import org.springframework.stereotype.Service; //Define las Variables a usar  
  
@Service  
public class CalculadoraPCFC extends CalculadoraDeFacturas {  
  
 //-- Propiedades  
 private BuscadorDeDatosPCFC buscadorDeDatos;  
 private YearMonth mesAFacturar;  
 private Plantas planta;  
  
 private List<BigDecimal> capacidadesPorEvento;  
  
 //-- Métodos Públicos  
  
 /\*\*  
 \* Para un mes dado y una fecha de pago dada, calcula si es necesario y devuelve los datos necesarios para crear una  
 \* Factura en un objeto <code>DatosPCFC</code>.  
 \*  
 \* @param mesAFacturar el mes para el cual se quieren los datos  
 \* @param fechaDePago fecha en la que se realizó el pago  
 \* @return datos para facturación  
 \*/  
 @Override  
 public DatosDeFacturacion getDatosParaFacturacion(Plantas planta, YearMonth mesAFacturar, Date fechaDePago, HashMap<String,BigDecimal> variablesManuales) {  
 setVariablesManuales(variablesManuales);  
 this.mesAFacturar = mesAFacturar;  
 this.planta = planta;  
  
 buscadorDeDatos = new BuscadorDeDatosPCFC(mesAFacturar, planta,variablesManuales);  
  
 datos = new DatosPCFC(); //Objeto que contiene los datos para la facturación  
  
 //Calcula las variables intermedias necesarias para realizar el cálculo final y las guarda en leerValorDeDato(Variables.  
 calcularVariablesIntermedias(fechaDePago);  
  
 //Calcula la fórmula general.  
  
 BigDecimal PCFCm = calcularPCFCm();  
 guardarValorDeDato(Variables.PCFCm, PCFCm);  
  
 guardarValorDeDato(Variables.monto, PCFCm);  
  
 String montoEnLetras = obtenerCantidadEnLetras((BigDecimal) leerValorDeDato(Variables.PCFCm));  
 guardarValorDeDato(Variables.MontoEnLetras, montoEnLetras);  
  
 return datos;  
 }  
  
 private BigDecimal obtenerPCFCm(BigDecimal cfcm, BigDecimal kc, BigDecimal faddm) {  
 return cfcm.multiply(kc).multiply(faddm);  
 }  
  
 private BigDecimal obtenerCFCam(BigDecimal cfcm, BigDecimal f) {  
 return cfcm.multiply(f.multiply(new BigDecimal(.8)).add(new BigDecimal(1)));  
 }  
  
  
 private BigDecimal obtenerHm(int mes, int año) {  
 return new BigDecimal(DateUtil.diasDelMes(mes, año)).multiply(new BigDecimal(24));  
 }  
  
 private BigDecimal obtenerYm(BigDecimal hm, List<BigDecimal> xm) {  
 double total = 0;  
 for (BigDecimal xmi : xm) {  
 total += xmi.doubleValue();  
 }  
 return hm.subtract(new BigDecimal(total));  
 }  
  
 private BigDecimal obtenerKc() {  
  
 guardarValorDeDato(Variables.KC, buscadorDeDatos.getKC());  
 return (BigDecimal) leerValorDeDato(Variables.KC);  
 }  
  
  
 private BigDecimal obtenerCfcm() {  
  
 guardarValorDeDato(Variables.CFCm, buscadorDeDatos.getCFCm(planta, mesAFacturar));  
 return (BigDecimal) leerValorDeDato(Variables.CFCm);  
 }  
  
 private List<BigDecimal> obtenerCfcmL(int n, int año) {  
 List<BigDecimal> lista = new ArrayList<>();  
 for (int i = 0; i < n; i++) {  
 Date date = new Date(año, i, 1);  
 YearMonth yearMonth = YearMonth.from(date.toInstant()  
 .atZone(ZoneId.systemDefault())  
 .toLocalDate());  
 try {  
 lista.add((BigDecimal) buscadorDeDatos.getCFCm(planta, mesAFacturar));  
  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 return lista;  
 }  
 }  
 return lista;  
 }  
  
 private BigDecimal obtenerT0() {  
  
 guardarValorDeDato(Variables.T0, buscadorDeDatos.getT0());  
 return (BigDecimal) leerValorDeDato(Variables.T0);  
 }  
  
 private BigDecimal obtenerT1() {  
  
 guardarValorDeDato(Variables.T1, buscadorDeDatos.getT1());  
 return (BigDecimal) leerValorDeDato(Variables.T1);  
 }  
  
 private BigDecimal obtenerPdgm() {  
  
  
 guardarValorDeDato(Variables.PDGm, buscadorDeDatos.getPDGm(mesAFacturar, planta));  
 return (BigDecimal) leerValorDeDato(Variables.PDGm);  
 }  
  
 private BigDecimal obtenerPdd(List<BigDecimal> fdedmL) {  
 BigDecimal sumaFdedm = new BigDecimal(0);  
 for (BigDecimal fdedm : fdedmL) {  
 sumaFdedm = sumaFdedm.add(fdedm);  
 }  
 return sumaFdedm.divide(new BigDecimal(12), 15, RoundingMode.HALF\_UP);  
 }  
  
 private BigDecimal obtenerPdd(Plantas planta, YearMonth mes) {  
  
 return calcularPDDm(planta, mes);  
 //return new BigDecimal(.974);  
 }  
  
 private BigDecimal obtenerFaddm(BigDecimal fadd, BigDecimal hm, BigDecimal ym, List<BigDecimal> xm,  
 List<BigDecimal> d, BigDecimal kc) {  
 BigDecimal a = fadd.divide(hm, 15, RoundingMode.HALF\_UP);  
  
 BigDecimal sumatoria = new BigDecimal(0);  
  
 int i = 0;  
 for (BigDecimal xmi : xm) {  
 sumatoria = sumatoria.add(xmi.multiply(d.get(i++).divide(kc)));  
 }  
 BigDecimal b = ym.add(sumatoria);  
 return a.multiply(b);  
 }  
  
  
 private BigDecimal obtenerF(List<BigDecimal> cfcmL, BigDecimal t0, BigDecimal t1) {  
 BigDecimal numerador = new BigDecimal(0);  
 BigDecimal denominador = new BigDecimal(0);  
 System.out.println("numero de cfcm:" + cfcmL.size());  
 int i = 1;  
 for (BigDecimal cfci : cfcmL) {  
 System.out.println("valor num:" + cfci.divide(t0.divide(new BigDecimal(1200), 15, RoundingMode.HALF\_UP)  
 .add(new BigDecimal(1))  
 .pow(i + 31), 15, RoundingMode.HALF\_UP));  
  
 System.out.println("valor den:" + cfci.divide(t1.divide(new BigDecimal(1200), 15, RoundingMode.HALF\_UP)  
 .add(new BigDecimal(1))  
 .pow(i + 31), 15, RoundingMode.HALF\_UP));  
  
 numerador = numerador.add(cfci.divide(t0.divide(new BigDecimal(1200), 15, RoundingMode.HALF\_UP)  
 .add(new BigDecimal(1))  
 .pow(i + 31), 15, RoundingMode.HALF\_UP));  
 denominador = denominador.add(cfci.divide(t1.divide(new BigDecimal(1200), 15, RoundingMode.HALF\_UP)  
 .add(new BigDecimal(1))  
 .pow(i + 31), 15, RoundingMode.HALF\_UP));  
  
 i++;  
  
 }  
 System.out.println(numerador.doubleValue());  
 System.out.println(denominador.doubleValue());  
 return numerador.divide(denominador, 15, RoundingMode.HALF\_UP);  
 }  
  
 private BigDecimal obtenerVmin(BigDecimal pdgm) {  
 return pdgm.multiply(new BigDecimal(.4924));  
 }  
  
 private BigDecimal obtenerFcor(BigDecimal pdgm) {  
 return new BigDecimal(1.97).divide(pdgm, 15, RoundingMode.HALF\_UP);  
 }  
  
 private BigDecimal obtenerFadd(BigDecimal vmin, BigDecimal pdd, BigDecimal pdgm, BigDecimal fcor) {  
 if (capacidadesPorEvento.size() == 0) {  
 return new BigDecimal(1);  
 } else {  
 if (pdgm.doubleValue() <= .96) {  
 if (pdd.doubleValue() >= 0 && pdd.doubleValue() <= vmin.doubleValue())  
 return new BigDecimal(0);  
 if (pdd.doubleValue() > vmin.doubleValue() && pdd.doubleValue() < pdgm.doubleValue())  
 return fcor.multiply(pdd).subtract(new BigDecimal(.97));  
 if (pdd.doubleValue() >= pdgm.doubleValue() && pdd.doubleValue() <= .96)  
 return new BigDecimal(1);  
 if (pdd.doubleValue() > .96 && pdd.doubleValue() < 1)  
 return pdd.multiply(new BigDecimal(1.5)).subtract(new BigDecimal(.44));  
 } else {  
  
 if (pdd.doubleValue() >= 0 && pdd.doubleValue() <= vmin.doubleValue())  
 return new BigDecimal(0);  
 if (pdd.doubleValue() > vmin.doubleValue() && pdd.doubleValue() < pdgm.doubleValue())  
 return fcor.multiply(pdd).subtract(new BigDecimal(.97));  
 if (pdd.doubleValue() > .96 && pdd.doubleValue() < 1)  
 return pdd.multiply(new BigDecimal(1.5)).subtract(new BigDecimal(.44));  
 }  
 }  
 return null;  
 }  
  
 private List<BigDecimal> obtenerD() {  
 return capacidadesPorEvento;  
 }  
  
 private List<BigDecimal> obtenerXm() {  
 //TODO: Obtener eventos falla  
 // List<TxEvento> eventosFalla = daoTxEvento.findByClasificacionEventoId\_ClasificacionEventoId(new BigDecimal(2));  
 List<TxEvento> eventosFalla = new ArrayList<>();  
 capacidadesPorEvento = new ArrayList<>();  
 List<BigDecimal> xms = new ArrayList<>();  
 for (TxEvento eventoFalla : eventosFalla) {  
 Period horas =  
 new Period(new DateTime(eventoFalla.getFechaInicio()), new DateTime(eventoFalla.getFechaFinal()));  
 double horasTotales = horas.getHours() + (horas.getMinutes() / 60) + (horas.getSeconds() / 3600);  
 xms.add(new BigDecimal(horasTotales));  
 capacidadesPorEvento.add(new BigDecimal(eventoFalla.getPotencia()));  
 }  
 return xms;  
 }  
  
 private BigDecimal obtenerCFCam(BigDecimal cprciom, BigDecimal cpdam, BigDecimal cidam) {  
  
 return cprciom.add(cpdam.add(cidam));  
 }  
  
 private BigDecimal obtenerCprciom() {  
  
 guardarValorDeDato(Variables.CPRCIOm, buscadorDeDatos.getCPRCIOm(planta, mesAFacturar));  
 return (BigDecimal) leerValorDeDato(Variables.CPRCIOm);  
 }  
  
 private BigDecimal obtenerF() {  
  
 guardarValorDeDato(Variables.FCPDam, buscadorDeDatos.getF\_CPDam(planta, mesAFacturar));  
 return (BigDecimal) leerValorDeDato(Variables.FCPDam);  
  
 }  
  
 private BigDecimal obtenerCpdm() {  
  
 guardarValorDeDato(Variables.CPDm, buscadorDeDatos.getCPDm(planta, mesAFacturar));  
 return (BigDecimal) leerValorDeDato(Variables.CPDm);  
 }  
  
 private BigDecimal obtenerCpdam(BigDecimal f, BigDecimal cpdm, BigDecimal t1, BigDecimal t0) {  
 return new BigDecimal(1).subtract(f)  
 .multiply(cpdm)  
 .add(t1.divide(t0, 15, RoundingMode.HALF\_UP).multiply(f.multiply(cpdm)));  
 }  
  
 private BigDecimal obtenerCidm() {  
  
 guardarValorDeDato(Variables.CIDm, buscadorDeDatos.getCIDm(planta, mesAFacturar));  
 return (BigDecimal) leerValorDeDato(Variables.CIDm);  
 }  
  
 private BigDecimal obtenerCidam(BigDecimal cpdam, BigDecimal t1, BigDecimal t0, BigDecimal cpdm, BigDecimal cidm) {  
 return t1.divide(t0, 15, RoundingMode.HALF\_UP).multiply(cpdam.divide(cpdm).multiply(cidm));  
 }  
  
 //-- Métodos Privados  
 private void calcularVariablesIntermedias(Date fechaDePago) {  
  
 }  
  
 private BigDecimal calcularPCFCm() {  
 int n = mesAFacturar.getMonthValue();  
 BigDecimal cfcam = new BigDecimal(0);  
 BigDecimal cfcm = new BigDecimal(0);  
  
 BigDecimal t0 =  
 obtenerT0(); //Rendimiento hasta el vencimiento en los Bonos de Tesoreria una fecha anterior a la fecha de presentación de la Propuesta.  
  
 guardarValorDeDato(Variables.T0, t0);  
 BigDecimal t1 =  
 obtenerT1(); //Rendimiento hasta el vencimiento en los Bonos de Tesoreria a una fecha de Inicio.  
  
 guardarValorDeDato(Variables.T1, t1);  
 if (planta.getID().intValue() == Globales.EAT) {  
 List<BigDecimal> cfcL =  
 obtenerCfcmL(n, mesAFacturar.getYear()); //Lista de Cargo fijo de capacidad por dia en el mes  
  
 BigDecimal f = obtenerF(cfcL, t0, t1); //  
 guardarValorDeDato(Variables.F, f);  
 cfcm = obtenerCfcm(); //Cargo Fijo de Capacidad para el mes m usd/kWm  
 guardarValorDeDato(Variables.CFCm, cfcm);  
  
 cfcam = obtenerCFCam(cfcm, f); //Cargo Fijo de Capacidad para el mes m ajuwstado usd/kWm  
 guardarValorDeDato(Variables.CFCam, cfcam);  
 } else {  
 if (planta.getID().intValue() == Globales.EST) {  
 BigDecimal cprciom = obtenerCprciom();  
 BigDecimal f = obtenerF();  
 guardarValorDeDato(Variables.F, f);  
 BigDecimal cpdm = obtenerCpdm();  
 guardarValorDeDato(Variables.CPDm, cpdm);  
  
 BigDecimal cpdam = obtenerCpdam(f, cpdm, t1, t0);  
  
 guardarValorDeDato(Variables.CPDam, cpdam);  
 BigDecimal cidm = obtenerCidm();  
 guardarValorDeDato(Variables.CIDm, cidm);  
  
 BigDecimal cidam = obtenerCidam(cpdam, t1, t0, cpdm, cidm);  
  
 guardarValorDeDato(Variables.CIDam, cidam);  
 cfcam = obtenerCFCam(cprciom, cpdam, cidam);  
 guardarValorDeDato(Variables.CFCam, cfcam);  
  
 }  
  
 }  
 BigDecimal kc = obtenerKc(); //Capacidad demostrada  
 guardarValorDeDato(Variables.KC, kc);  
 BigDecimal hm = obtenerHm(mesAFacturar.getMonthValue(), mesAFacturar.getYear()); //Numero de horas en el mes  
 guardarValorDeDato(Variables.Hm, hm);  
  
 List<BigDecimal> xm =  
 obtenerXm(); //Lista de numero de horas por evento en el mes m en las cuales hubo evento que afecte la disponibilidad de la capacidad demostrada KC  
 BigDecimal ym = obtenerYm(hm, xm); //Numero de horas en el mes m en las cuales no hubo evento  
 List<BigDecimal> d = obtenerD(); //Lista de capacidades declaradas disponibles por evento  
 BigDecimal pdgm = obtenerPdgm();  
 BigDecimal vmin = obtenerVmin(pdgm);  
 BigDecimal fcor = obtenerFcor(pdgm);  
 //List<BigDecimal> fdedmL = obtenerFdedmL(dom.getFechaFactura());  
  
 BigDecimal pdd = obtenerPdd(planta, mesAFacturar);  
  
 guardarValorDeDato(Variables.PDDm, pdd);  
 BigDecimal fadd = obtenerFadd(vmin, pdd, pdgm, fcor); //Factor de Ajuste por Disponibilidad Demostrada  
 guardarValorDeDato(Variables.FADD, fadd);  
  
 BigDecimal faddm =  
 obtenerFaddm(fadd, hm, ym, xm, d, kc); //Factor de Ajuste por Disponibilidad Demostrada en el mes  
 guardarValorDeDato(Variables.FADDm, faddm);  
  
  
 BigDecimal pcfcm = obtenerPCFCm(cfcam, kc, faddm); //Pago del cargo fijo de capacidad en el mes m  
 return pcfcm;  
 }  
  
 private BigDecimal calcularFADDm() {  
 //final int precisionCapacidad = 6;  
 //final int precisionFADDm = 16;  
 BigDecimal sumatorio = BigDecimal.ZERO;  
 BigDecimal PDDm = calcularPDDm(planta, mesAFacturar);  
 BigDecimal PDGm = (BigDecimal) buscadorDeDatos.getPDGm(mesAFacturar, planta);  
  
 guardarValorDeDato(Variables.FDEDm, calcularFDEDm(planta, mesAFacturar));  
 guardarValorDeDato(Variables.PDDm, PDDm);  
 guardarValorDeDato(Variables.PDGm, PDGm);  
  
 guardarValorDeDato(Variables.FADD, calcularFADD(PDDm, PDGm));  
 guardarValorDeDato(Variables.Hm, buscadorDeDatos.getHorasDelMes(mesAFacturar));  
 guardarValorDeDato(Variables.Xm, BigDecimal.ZERO);  
  
 //Suma todos los Xm\*D'/KC de los eventos que afectaron la disponibilidad  
 ArrayList<EventosDeDisponibilidad> eventosDeDisponibilidad = buscadorDeDatos.getEventosDeDisponibilidad();  
 if (eventosDeDisponibilidad != null) {  
 for (EventosDeDisponibilidad evento : eventosDeDisponibilidad) {  
 BigDecimal Xm = (BigDecimal) leerValorDeDato(Variables.Xm);  
 BigDecimal KC = (BigDecimal) leerValorDeDato(Variables.KC);  
  
 guardarValorDeDato(Variables.Xm, Xm.add(evento.getDuracion()));  
 //BigDecimal factorDeCapacidad = evento.getDuracion().multiply(evento.getCapacidadDeclarada().divide(leerValorDeDato(Variables.KC, precisionCapacidad, RoundingMode.HALF\_DOWN));  
 BigDecimal factorDeCapacidad =  
 evento.getDuracion().multiply(evento.getCapacidadDeclarada().divide(KC, RoundingMode.HALF\_DOWN));  
 sumatorio = sumatorio.add(factorDeCapacidad);  
 }  
 } else {  
 //SI no hay eventos, Ym = Hm  
 System.out.print("No hay eventos que afecten la disponibilidad, FADDm = 1\n");  
 guardarValorDeDato(Variables.Ym, leerValorDeDato(Variables.Hm));  
 return BigDecimal.ONE;  
 }  
  
 //Ym = Hm - Xm  
 BigDecimal Hm = (BigDecimal) leerValorDeDato(Variables.Hm);  
 BigDecimal Xm = (BigDecimal) leerValorDeDato(Variables.Xm);  
 BigDecimal Ym = Hm.subtract(Xm);  
  
 guardarValorDeDato(Variables.Ym, Ym);  
 //BigDecimal FADDm = leerValorDeDato(Variables.FADD.divide(leerValorDeDato(Variables.Hm, precisionFADDm, RoundingMode.HALF\_DOWN);  
 BigDecimal FADD = (BigDecimal) leerValorDeDato(Variables.FADD);  
 BigDecimal FADDm = FADD.divide(Hm, RoundingMode.HALF\_DOWN);  
 FADDm = FADDm.multiply(Ym.add(sumatorio));  
 return FADDm;  
 }  
  
 private BigDecimal calcularFADD(BigDecimal PDDm, BigDecimal PDGm) {  
 BigDecimal FADD;  
 BigDecimal Vmin = PDGm.multiply(new BigDecimal("0.4924"));  
 BigDecimal Fcor = new BigDecimal("1.97").divide(PDGm, 3, RoundingMode.HALF\_DOWN);  
 if (!(PDDm.compareTo(BigDecimal.ZERO) >= 0)) {  
 System.out.printf("Valor de PPDm invalido para el mes %s\n", mesAFacturar.toString());  
 FADD = null;  
 } else {  
 if (PDDm.compareTo(Vmin) <= 0) {  
 FADD = BigDecimal.ZERO;  
 } else if (PDDm.compareTo(PDGm) < 0) {  
 FADD = Fcor.multiply(PDDm).subtract(new BigDecimal("0.97"));  
 } else {  
 if (PDDm.compareTo(new BigDecimal("0.960")) <= 0) {  
 FADD = BigDecimal.ONE;  
 } else { //Si PDD > 0.960  
 FADD = PDDm.multiply(new BigDecimal("1.5")).subtract(new BigDecimal("0.44"));  
 }  
 }  
 }  
 return FADD;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\calculadoradefacturas\CalculadoraPCFOM.java

package mx.mbi.dgc.business.facturacion.facturador.calculadoradefacturas;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCFOM;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCFOM.Variables; //Define las Variables a usar  
  
import mx.mbi.dgc.business.facturacion.facturador.EventosDeDisponibilidad;  
import mx.mbi.dgc.business.facturacion.facturador.buscadordedatos.BuscadorDeDatosPCFOM;  
import mx.mbi.dgc.dao.negocio.facturacion.DAOFacturacion;  
import mx.mbi.dgc.global.Globales.Plantas;  
  
import java.math.BigDecimal;  
import java.math.RoundingMode;  
import java.time.YearMonth;  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.HashMap;  
  
public class CalculadoraPCFOM extends CalculadoraDeFacturas {  
  
 //-- Propiedades  
 private BuscadorDeDatosPCFOM buscadorDeDatos;  
 private YearMonth mesAFacturar;  
 private Plantas planta;  
  
  
 //-- Métodos Públicos  
 /\*\*  
 \* Para un mes dado y una fecha de pago dada, calcula si es necesario y devuelve los datos necesarios para crear una  
 \* Factura en un objeto <code>DatosPCFOM</code>.  
 \*  
 \* @param mesAFacturar el mes para el cual se quieren los datos  
 \* @param fechaDePago fecha en la que se realizó el pago  
 \* @return datos para facturación  
 \*/  
 @Override  
 public DatosDeFacturacion getDatosParaFacturacion(Plantas planta, YearMonth mesAFacturar, Date fechaDePago, HashMap<String,BigDecimal> variablesManuales) {  
 setVariablesManuales(variablesManuales);  
 this.mesAFacturar = mesAFacturar;  
 this.planta = planta;  
  
 buscadorDeDatos = new BuscadorDeDatosPCFOM(mesAFacturar, planta,variablesManuales);  
 super.buscadorDeDatos.setVariablesManuales(variablesManuales);  
 datos = new DatosPCFOM(); //Objeto que contiene los datos para la facturación  
  
 //Calcula las variables intermedias necesarias para realizar el cálculo final y las guarda en leerValorDeDato(Variables.  
 System.out.println(variablesManuales);  
 System.out.println(buscadorDeDatos.getVariablesManuales());  
 System.out.println(super.buscadorDeDatos.getVariablesManuales());  
 calcularVariablesIntermedias(fechaDePago);  
  
 //Calcula la fórmula general.  
 BigDecimal PCFOMm=calcularPCFOMm();  
 guardarValorDeDato(Variables.PCFOMm,PCFOMm );  
 guardarValorDeDato(Variables.monto,PCFOMm );  
  
 BigDecimal montoFinal = (BigDecimal)leerValorDeDato(Variables.PCFOMm);  
 String montoEnLetras = obtenerCantidadEnLetras(montoFinal.multiply(new BigDecimal("0.16")));  
 guardarValorDeDato(Variables.MontoEnLetras, montoEnLetras);  
  
 return datos;  
 }  
  
  
 //-- Métodos Privados  
 private void calcularVariablesIntermedias(Date fechaDePago) {  
 DAOFacturacion daoFacturacion = new DAOFacturacion();  
 System.out.printf("Calculando variables intermedias para %s\n", fechaDePago.toString());  
 calcularINPPmo(mesAFacturar); //Guarda INPPIm, INPPIo, y calcula y guarda INPPImo  
 guardarValorDeDato(Variables.CFOMmtm, buscadorDeDatos.getCFOMmtm());  
  
 calcularUSPPImo(); //Guarda USPPIm, USPPIo y calcula y guarda USPPImo  
 guardarValorDeDato(Variables.TCp, buscadorDeDatos.getTCp(fechaDePago));  
 BigDecimal CFOMdm = buscadorDeDatos.getCFOMdm();  
 daoFacturacion.setCFOMdm(planta, mesAFacturar, CFOMdm);  
 guardarValorDeDato(Variables.CFOMdm, CFOMdm);  
  
 guardarValorDeDato(Variables.IESm, buscadorDeDatos.getIESm());  
 guardarValorDeDato(Variables.CFOMmom, buscadorDeDatos.getCFOMmom());  
  
 guardarValorDeDato(Variables.KC, buscadorDeDatos.getKC());  
 guardarValorDeDato(Variables.FADDm, calcularFADDm());  
 }  
  
 private BigDecimal calcularPCFOMm() {  
 BigDecimal CFOMmtm = (BigDecimal) leerValorDeDato(Variables.CFOMmtm);  
 BigDecimal INPP = (BigDecimal) leerValorDeDato(Variables.INPP);  
 BigDecimal PCFOMm = CFOMmtm.multiply(INPP);  
  
 BigDecimal CFOMdm = (BigDecimal) leerValorDeDato(Variables.CFOMdm);  
 BigDecimal USPPI = (BigDecimal) leerValorDeDato(Variables.USPPI);  
 BigDecimal TCp = (BigDecimal) leerValorDeDato(Variables.TCp);  
 PCFOMm = PCFOMm.add(  
 CFOMdm.multiply(USPPI).multiply(TCp));  
  
 BigDecimal CFOMmom = (BigDecimal) leerValorDeDato(Variables.CFOMmom);  
 BigDecimal IESm = (BigDecimal) leerValorDeDato(Variables.IESm);  
 PCFOMm = PCFOMm.add(CFOMmom.multiply(IESm));  
  
 BigDecimal KC = (BigDecimal) leerValorDeDato(Variables.KC);  
 BigDecimal FADDm = (BigDecimal) leerValorDeDato(Variables.FADDm);  
 PCFOMm = PCFOMm.multiply(KC).multiply(FADDm);  
 return PCFOMm;  
 }  
  
 private void calcularUSPPImo() {  
 final int precisionUSPPI = 6;  
 BigDecimal USPPIm = buscadorDeDatos.getUSPPIm(mesAFacturar);  
 BigDecimal USPPIo = buscadorDeDatos.getUSPPIo();  
 guardarValorDeDato(Variables.USPPIm, USPPIm);  
 guardarValorDeDato(Variables.USPPIo, USPPIo);  
  
 guardarValorDeDato(Variables.USPPI, USPPIm.divide(USPPIo, precisionUSPPI, RoundingMode.HALF\_DOWN));  
 }  
  
 private BigDecimal calcularFADDm() {  
 //final int precisionCapacidad = 6;  
 //final int precisionFADDm = 16;  
 BigDecimal sumatorio = BigDecimal.ZERO;  
 BigDecimal PDDm = calcularPDDm(planta, mesAFacturar);  
 BigDecimal PDGm = buscadorDeDatos.getPDGm();  
  
 guardarValorDeDato(Variables.FDEDm, calcularFDEDm(planta, mesAFacturar));  
 guardarValorDeDato(Variables.PDDm, PDDm);  
 guardarValorDeDato(Variables.PDGm, PDGm);  
  
 guardarValorDeDato(Variables.FADD, calcularFADD(PDDm, PDGm));  
 guardarValorDeDato(Variables.Hm, buscadorDeDatos.getHorasDelMes(mesAFacturar));  
 guardarValorDeDato(Variables.Xm, BigDecimal.ZERO);  
  
 //Suma todos los Xm\*D'/KC de los eventos que afectaron la disponibilidad  
 ArrayList<EventosDeDisponibilidad> eventosDeDisponibilidad = buscadorDeDatos.getEventosDeDisponibilidad();  
 if (eventosDeDisponibilidad != null) {  
 for (EventosDeDisponibilidad evento : eventosDeDisponibilidad) {  
 BigDecimal Xm = (BigDecimal) leerValorDeDato(Variables.Xm);  
 BigDecimal KC = (BigDecimal) leerValorDeDato(Variables.KC);  
  
 guardarValorDeDato(Variables.Xm, Xm.add(evento.getDuracion()));  
 //BigDecimal factorDeCapacidad = evento.getDuracion().multiply(evento.getCapacidadDeclarada().divide(leerValorDeDato(Variables.KC, precisionCapacidad, RoundingMode.HALF\_DOWN));  
 BigDecimal factorDeCapacidad = evento.getDuracion().multiply(  
 evento.getCapacidadDeclarada().divide(KC, RoundingMode.HALF\_DOWN));  
 sumatorio = sumatorio.add(factorDeCapacidad);  
 }  
 } else {  
 //SI no hay eventos, Ym = Hm  
 System.out.print("No hay eventos que afecten la disponibilidad, FADDm = 1\n");  
 guardarValorDeDato(Variables.Ym, leerValorDeDato(Variables.Hm));  
 return BigDecimal.ONE;  
 }  
  
 //Ym = Hm - Xm  
 BigDecimal Hm = (BigDecimal) leerValorDeDato(Variables.Hm);  
 BigDecimal Xm = (BigDecimal) leerValorDeDato(Variables.Xm);  
 BigDecimal Ym = Hm.subtract(Xm);  
  
 guardarValorDeDato(Variables.Ym, Ym);  
 //BigDecimal FADDm = leerValorDeDato(Variables.FADD.divide(leerValorDeDato(Variables.Hm, precisionFADDm, RoundingMode.HALF\_DOWN);  
 BigDecimal FADD = (BigDecimal) leerValorDeDato(Variables.FADD);  
 BigDecimal FADDm = FADD.divide(Hm, RoundingMode.HALF\_DOWN);  
 FADDm = FADDm.multiply(Ym.add(sumatorio));  
 return FADDm;  
 }  
  
 private BigDecimal calcularFADD(BigDecimal PDDm, BigDecimal PDGm) {  
 BigDecimal FADD;  
 BigDecimal Vmin = PDGm.multiply(new BigDecimal("0.4924"));  
 BigDecimal Fcor = new BigDecimal("1.97").divide(PDGm, 3, RoundingMode.HALF\_DOWN);  
 if (!(PDDm.compareTo(BigDecimal.ZERO) >= 0)) {  
 System.out.printf("Valor de PPDm invalido para el mes %s\n", mesAFacturar.toString());  
 FADD = null;  
 } else {  
 if (PDDm.compareTo(Vmin) <= 0) {  
 FADD = BigDecimal.ZERO;  
 } else if (PDDm.compareTo(PDGm) < 0) {  
 FADD = Fcor.multiply(PDDm).subtract(new BigDecimal("0.97"));  
 } else {  
 if (PDDm.compareTo(new BigDecimal("0.960")) <= 0) {  
 FADD = BigDecimal.ONE;  
 } else { //Si PDD > 0.960  
 FADD = PDDm.multiply(new BigDecimal("1.5")).subtract(new BigDecimal("0.44"));  
 }  
 }  
 }  
 return FADD;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\calculadoradefacturas\CalculadoraPCVOM.java

package mx.mbi.dgc.business.facturacion.facturador.calculadoradefacturas;  
  
import java.math.BigDecimal;  
  
import java.time.YearMonth;  
  
import java.util.Date;  
import java.util.HashMap;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCFOM;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCVOM;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosPCVOM.Variables;  
import mx.mbi.dgc.business.facturacion.facturador.buscadordedatos.BuscadorDeDatosPCVOM;  
import mx.mbi.dgc.global.Globales.Plantas; //Define las Variables a usar  
  
public class CalculadoraPCVOM extends CalculadoraDeFacturas {  
  
 private BuscadorDeDatosPCVOM buscadorDeDatos;  
 private YearMonth mesAFacturar;  
 private Plantas planta;  
   
  
 /\*\*  
 \* Para un mes dado y una fecha de pago dada, calcula si es necesario y devuelve los datos necesarios para crear una  
 \* Factura en un objeto <code>DatosPCVOM</code>.  
 \*  
 \* @param mesAFacturar el mes para el cual se quieren los datos  
 \* @param fechaDePago fecha en la que se realizó el pago  
 \* @return datos para facturación  
 \*/  
 @Override  
 public DatosDeFacturacion getDatosParaFacturacion(Plantas planta, YearMonth mesAFacturar, Date fechaDePago, HashMap<String,BigDecimal> variablesManuales) {  
 setVariablesManuales(variablesManuales);  
 this.mesAFacturar = mesAFacturar;  
 this.planta = planta;  
  
 buscadorDeDatos = new BuscadorDeDatosPCVOM(mesAFacturar, planta,variablesManuales);  
  
 super.buscadorDeDatos.setVariablesManuales(variablesManuales);  
 datos = new DatosPCVOM(); //Objeto que contiene los datos para la facturación  
  
 //Calcula las variables intermedias necesarias para realizar el cálculo final y las guarda en leerValorDeDato(Variables.  
 calcularVariablesIntermedias(fechaDePago);  
  
 //Calcula la fórmula general.  
 BigDecimal PCVOMm=calcularPCVOMm();  
 guardarValorDeDato(Variables.PCVOMm, PCVOMm);  
 guardarValorDeDato(Variables.monto, PCVOMm);  
  
  
 BigDecimal montoFinal = (BigDecimal)leerValorDeDato(Variables.PCVOMm);  
 String montoEnLetras = obtenerCantidadEnLetras(montoFinal.multiply(new BigDecimal("0.16")));  
 guardarValorDeDato(DatosPCFOM.Variables.MontoEnLetras, montoEnLetras);  
  
 return datos;  
 }  
  
  
 private void calcularVariablesIntermedias(Date fechaDePago) {  
 System.out.printf("Calculando variables intermedias para %s\n", fechaDePago.toString());  
  
 guardarValorDeDato(Variables.CVOMm, buscadorDeDatos.getCVOMm(fechaDePago));  
 calcularINPPmo(mesAFacturar); //Guarda INPPIm, INPPIo, y calcula y guarda INPPImo  
 guardarValorDeDato(Variables.PCAm, buscadorDeDatos.getPCAm(fechaDePago));  
 guardarValorDeDato(Variables.CCAm, buscadorDeDatos.getCCAm(fechaDePago));  
 guardarValorDeDato(Variables.sumatoriaEh, calcularSmatoriaEh(fechaDePago));  
  
  
 }  
  
 private BigDecimal calcularPCVOMm() {  
 BigDecimal CVOMm = (BigDecimal) leerValorDeDato(Variables.CVOMm);  
 BigDecimal INPP = (BigDecimal) leerValorDeDato(DatosPCFOM.Variables.INPP);  
 BigDecimal PCAm = (BigDecimal) leerValorDeDato(Variables.PCAm);  
 BigDecimal CCAm = (BigDecimal) leerValorDeDato(Variables.CCAm);  
 BigDecimal sumatoriaEh = (BigDecimal) leerValorDeDato(Variables.sumatoriaEh);  
  
 return CVOMm.multiply(INPP.add(PCAm.multiply(CCAm)).multiply(sumatoriaEh));  
 }  
  
 /\*private void calcularINPPmo() {  
 final int precisionINPP = 14;  
 guardarValorDeDato(Variables.INPPm, buscadorDeDatos.getINPPm());  
 guardarValorDeDato(Variables.INPPo, buscadorDeDatos.getINPPo());  
 guardarValorDeDato(Variables.INPP,  
 leerValorDeDato(Variables.INPPm)  
 .divide(leerValorDeDato(Variables.INPPo), precisionINPP, RoundingMode.HALF\_DOWN));  
 }\*/  
  
 private BigDecimal calcularSmatoriaEh(Date fecha) {  
 //TODO: realizar calculos para SmatoriaEh  
   
  
 return new BigDecimal("121.72594174551");  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\datosdefacturacion\DatosAOMXLT.java

package mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion;  
  
  
import mx.mbi.dgc.business.facturacion.facturador.generadordedocumentos.GeneradorDeDocumentosDeFacturacion;  
  
public class DatosAOMXLT extends DatosDeFacturacion {  
 public enum Variables implements DatosDeFacturacion.Variables{  
 AOMXLT("AOMXLT"),  
 MontoEnLetras("MontoEnLetras"),  
 CFOMdm("CFOMdm"),  
 KC("KC"),  
 USPPIm("USPPIm"),  
 USPPIo("USPPIo"),  
 USPPImo("USPPImo"),  
 TCp("TCp"),  
 AX("AX"),  
 monto("monto"),  
 HX("HX");  
  
  
 final String nombre;  
  
 Variables(String nombre) {  
 this.nombre = nombre;  
 }  
  
 @Override  
 public String getSimbolo() {  
 return GeneradorDeDocumentosDeFacturacion.corchetes[0] + nombre + GeneradorDeDocumentosDeFacturacion.corchetes[1];  
 }  
  
 @Override  
 public String getNombre() {  
 return nombre;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\datosdefacturacion\DatosDeFacturacion.java

package mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion;  
  
import java.math.BigDecimal;  
import java.util.ArrayList;  
import java.util.HashMap;  
  
public abstract class DatosDeFacturacion {  
 public interface Variables{  
 String getNombre();  
  
 String getSimbolo();  
  
 }  
  
 public HashMap<Variables, ArrayList<Object>> valores = new HashMap<>();  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\datosdefacturacion\DatosNulos.java

package mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion;  
  
public class DatosNulos extends DatosDeFacturacion {  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\datosdefacturacion\DatosPCA.java

package mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion;  
  
  
import mx.mbi.dgc.business.facturacion.facturador.generadordedocumentos.GeneradorDeDocumentosDeFacturacion;  
  
public class DatosPCA extends DatosDeFacturacion {  
 public enum Variables implements DatosDeFacturacion.Variables {  
 PCAm("PCAm"),  
 PCAGm("PCAGm"),  
 PCAGmd("PCAGmd"),  
 CTPAgmd("CTPAgmd"),  
 PCADm("PCADm"),  
 PCSm("PCSm"),  
 PCIm("PCIm"),  
 PPGm("PPGm"),  
 PPGmd("PPGmd"),  
 CTPAgm("CTPAgm"),  
 CTRAgx("CTRAgx"),  
 NAgmx("NAgmx"),  
 NAgmdx("NAgmdx"),  
 CTPAdm("CTPAdm"),  
 PVDm("PVDm"),  
 PCSdm("PCSdm"),  
 PCIdm("PCIdm"),  
 PCSmd("PCSmd"),  
 PCImd("PCImd"),  
 CTRAdx("CTRAdx"),  
 NAdmx("NAdmx"),  
 monto("monto");  
 final String nombre;  
  
 Variables(String nombre) {  
 this.nombre = nombre;  
 }  
  
 @Override  
 public String getSimbolo() {  
 return GeneradorDeDocumentosDeFacturacion.corchetes[0] + nombre + GeneradorDeDocumentosDeFacturacion.corchetes[1];  
 }  
  
 @Override  
 public String getNombre() {  
 return nombre;  
 }  
  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\datosdefacturacion\DatosPCC.java

package mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion;  
  
  
import mx.mbi.dgc.business.facturacion.facturador.generadordedocumentos.GeneradorDeDocumentosDeFacturacion;  
  
public class DatosPCC extends DatosDeFacturacion {  
 public enum Variables implements DatosDeFacturacion.Variables {  
 PCCm("PCCm"),  
 MontoEnLetras("MontoEnLetras"),  
 PGm("PGm"),  
 PDm("PDm"),  
 PVDm("PVDm"),  
 PCIDm("PCIDm"),  
 PCSDm("PCSDm"),  
 CTPdm("CTPdm"),  
 CAm("CAm"),  
 PPGm("PPGm"),  
 PCImh("PCImh"),  
 PCSmh("PCSmh"),  
 CTPgmh("PCSmh"),  
 NHOVgmh("NHOVgmh"),  
 NHOVdm("NHOVdm"),  
 CTOVg("CTOVg"),  
 CTRghCSh("CTRghCSh"),  
 Egh("Egh"),  
 monto("monto");  
  
 final String nombre;  
  
 Variables(String nombre) {  
 this.nombre = nombre;  
 }  
  
 @Override  
 public String getSimbolo() {  
 return GeneradorDeDocumentosDeFacturacion.corchetes[0] + nombre + GeneradorDeDocumentosDeFacturacion.corchetes[1];  
 }  
  
 @Override  
 public String getNombre() {  
 return nombre;  
 }  
  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\datosdefacturacion\DatosPCFC.java

package mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion;  
  
  
import mx.mbi.dgc.business.facturacion.facturador.generadordedocumentos.GeneradorDeDocumentosDeFacturacion;  
  
public class DatosPCFC extends DatosDeFacturacion {  
 public enum Variables implements DatosDeFacturacion.Variables {  
 PCFCm("PCFCm"),  
 MontoEnLetras("MontoEnLetras"),  
 CFCm("CFCm"),  
 CFCdm("CFCdm"),  
 CFCam("CFCam"),  
 CPRCIOm("CPRCIOm"),  
 CPDm("CPDm"),  
 PDGm("PDGm"),  
 FCPDam("FCPDam"),  
 Hm("Hm"),  
 PDDm("PDDm"),  
 CIDm("CIDm"),  
 CPDam("CPDm"),  
 CIDam("CIDm"),  
 FADD("FADD"),  
 FADDm("FADDm"),  
 FDEDm("FDEDm"),  
 T0("T0"),  
 T1("T1"),  
 KC("KC"),  
 F("F"),  
 Xm("Xm"),  
 Ym("Ym"),  
 monto("monto");  
  
  
 final String nombre;  
  
 Variables(String nombre) {  
 this.nombre = nombre;  
 }  
  
 @Override  
 public String getSimbolo() {  
 return GeneradorDeDocumentosDeFacturacion.corchetes[0] + nombre + GeneradorDeDocumentosDeFacturacion.corchetes[1];  
 }  
  
 @Override  
 public String getNombre() {  
 return nombre;  
 }  
  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\datosdefacturacion\DatosPCFOM.java

package mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion;  
  
  
import mx.mbi.dgc.business.facturacion.facturador.generadordedocumentos.GeneradorDeDocumentosDeFacturacion;  
  
public class DatosPCFOM extends DatosDeFacturacion {  
 public enum Variables implements DatosDeFacturacion.Variables {  
 PCFOMm("PCFOMm"),  
 MontoEnLetras("MontoEnLetras"),  
 CFOMmtm("CFOMmtm"),  
 CFOMdm("CFOMdm"),  
 CFOMmom("CFOMmom"),  
 FADD("FADD"),  
 FADDm("FADDm"),  
 FDEDm("FDEDm"),  
 Hm("Hm"),  
 IESm("IESm"),  
 INPPo("INPPo"),  
 INPPm("INPPm"),  
 INPP("INPP"),  
 KC("KC"),  
 PDDm("PDDm"),  
 PDGm("PDGm"),  
 TCp("TCp"),  
 USPPIo("USPPIo"),  
 USPPIm("USPPIm"),  
 USPPI("USPPI"),  
 Xm("Xm"),  
 Ym("Ym"),  
 monto("monto");  
  
  
 final String nombre;  
  
 Variables(String nombre) {  
 this.nombre = nombre;  
 }  
  
 @Override  
 public String getSimbolo() {  
 return GeneradorDeDocumentosDeFacturacion.corchetes[0] + nombre + GeneradorDeDocumentosDeFacturacion.corchetes[1];  
 }  
  
 @Override  
 public String getNombre() {  
 return nombre;  
 }  
  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\datosdefacturacion\DatosPCVOM.java

package mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion;  
  
  
import mx.mbi.dgc.business.facturacion.facturador.generadordedocumentos.GeneradorDeDocumentosDeFacturacion;  
  
public class DatosPCVOM extends DatosDeFacturacion {  
 public enum Variables implements DatosDeFacturacion.Variables {  
 PCVOMm("PCVOMm"),  
 MontoEnLetras("MontoEnLetras"),  
 CVOMm("CVOMm"),  
 INPPo("INPPo"),  
 INPPm("INPPm"),  
 PCAm("PCAm"),  
 CCAm("CCAm"),  
 sumatoriaEh("sumatoriaEh"),  
 CVOMmtm("CVOMmtm"),  
 CVOMmom("CVOMmom"),  
 FADD("FADD"),  
 FADDm("FADDm"),  
 FDEDm("FDEDm"),  
 Hm("Hm"),  
 IESm("IESm"),  
 INPP("INPP"),  
 KC("KC"),  
 PDDm("PDDm"),  
 PDGm("PDGm"),  
 TCp("TCp"),  
 USPPIo("USPPIo"),  
 USPPIm("USPPIm"),  
 USPPI("USPPI"),  
 Xm("Xm"),  
 Ym("Ym"),  
 monto("monto");  
  
  
 final String nombre;  
  
 Variables(String nombre) {  
 this.nombre = nombre;  
 }  
  
 @Override  
 public String getSimbolo() {  
 return GeneradorDeDocumentosDeFacturacion.corchetes[0] + nombre + GeneradorDeDocumentosDeFacturacion.corchetes[1];  
 }  
  
 @Override  
 public String getNombre() {  
 return nombre;  
 }  
  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\EventosDeDisponibilidad.java

package mx.mbi.dgc.business.facturacion.facturador;  
  
import java.math.BigDecimal;  
  
public class EventosDeDisponibilidad {  
 private BigDecimal duracion;  
 private BigDecimal capacidadDeclarada;  
  
 EventosDeDisponibilidad(BigDecimal duracion, BigDecimal capacidadDeclarada) {  
 this.duracion = duracion;  
 this.capacidadDeclarada = capacidadDeclarada;  
 }  
  
 public BigDecimal getCapacidadDeclarada() {  
 return capacidadDeclarada;  
 }  
  
 public BigDecimal getDuracion() {  
 return duracion;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\Facturador.java

package mx.mbi.dgc.business.facturacion.facturador;  
  
import java.math.BigDecimal;  
  
import java.time.YearMonth;  
  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.Map;  
  
import mx.mbi.dgc.business.facturacion.facturador.calculadoradefacturas.\*;  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.business.facturacion.facturador.generadordedocumentos.\*;  
import mx.mbi.dgc.global.Globales;  
  
import org.springframework.stereotype.Service;  
  
  
/\*\*  
 \* Se encarga de crear las facturas usando una calculadora de facturas y un generador de documentos de facturación.  
 \* La calculadora y el generador de documentos se eligen dinámicamente según el TipoDeFactura.  
 \*/  
@Service  
public class Facturador {  
 //Para mayor reusabilidad del código y hacerlo más fácilmente extensible, permite usar un patrón de diseño Strategy  
  
  
 public enum TipoDeFactura {  
 /\*\*  
 \* Cuando no se reconoce un tipo de factura, es inválido,  
 \* con funcionalidad nula para evitar tener que estar revisando valores null.  
 \*/  
 INVALIDO(new CalculadoraINVALIDO(), new GeneradorDeDocumentosINVALIDO(), "INVALIDO"),  
  
 /\*\*  
 \* Pago por Cargo Fijo de Operación y Mantenimiento  
 \*/  
  
 PCFOM(new CalculadoraPCFOM(), new GeneradorDeDocumentosPCFOM(), "PCFOM"),  
  
 /\*\*  
 \* Pago por Cargo Fijo de Capacidad  
 \*/  
 PCFC(new CalculadoraPCFC(), new GeneradorDeDocumentosPCFC(), "PCFC"),  
  
 /\*\*  
 \* Pago por Cargo Variable de Operación y Mantenimiento  
 \*/  
 PCVOM(new CalculadoraPCVOM(), new GeneradorDeDocumentosPCVOM(), "PCVOM"),  
  
 /\*\*  
 \* Pago por Cargo por Combustible  
 \*/  
 PCCm(new CalculadoraPCC(), new GeneradorDeDocumentosPCC(), "PCC"),  
 /\*\*  
 \* Pago por Ajuste en el Cargo Fijo de Operación y Mantenimiento por Operación en Exceso de los Límites Técnicos.  
 \*/  
 AOMXLT(new CalculadoraAOMXLT(), new GeneradorDeDocumentosAOMXLT(), "AOMXLT"),  
 /\*\*  
 \* Pago por Cargo por Arranque  
 \*/  
 PCA(new CalculadoraPCA(), new GeneradorDeDocumentosPCA(), "PCA");  
  
 final CalculadoraDeFacturas calculadora;  
 final GeneradorDeDocumentosDeFacturacion generadorDeDocumentos;  
 private final String nombre;  
  
 TipoDeFactura(CalculadoraDeFacturas calculadora, GeneradorDeDocumentosDeFacturacion generadorDeDocumentos,  
 String nombre) {  
 this.calculadora = calculadora;  
 this.generadorDeDocumentos = generadorDeDocumentos;  
 this.nombre = nombre;  
 }  
  
 /\*\*  
 \* Para obtener el nombre del tipo de factura  
 \* @return el nombre  
 \*/  
 public String getNombre() {  
 return nombre;  
 }  
 }  
  
 //-- Atributos  
 private TipoDeFactura tipoDeFactura;  
 private BigDecimal monto;  
 private HashMap<String,BigDecimal> variableManuales;  
  
 private BigDecimal buscarMontoEnDatos(DatosDeFacturacion datosDeFacturacion) {  
 for (Map.Entry<DatosDeFacturacion.Variables, ArrayList<Object>> variable :  
 datosDeFacturacion.valores.entrySet()) {  
  
 if (variable.getKey()  
 .getNombre()  
 .equals("monto")) {  
 return (BigDecimal) variable.getValue().get(variable.getValue().size() - 1);  
 }  
 }  
 return null;  
 }  
  
 public void setMonto(BigDecimal monto) {  
 this.monto = monto;  
 }  
  
 public BigDecimal getMonto() {  
 return monto;  
 }  
 //-- Constructores  
 private Facturador() {  
 } //No se puede instanciar sin definir un tipo de pathFactura  
  
 /\*\*  
 \* Se inicializa definiendo el tipo de factura a crear.  
 \* @param tipoDeFactura el tipo de factura a crear  
 \* @param variablesManuales variables enviadas desde el front para establecer manualmente en el cálculo de facturación  
 \*/  
 public Facturador(TipoDeFactura tipoDeFactura,HashMap<String,BigDecimal> variablesManuales) { //Se debe inicializar con un tipo de pathFactura definido  
  
 System.out.printf("Creando factura: %s\n", tipoDeFactura.getNombre());  
 this.tipoDeFactura = tipoDeFactura;  
 this.variableManuales=variablesManuales;  
 }  
  
 //-- Métodos públicos  
  
 /\*\*  
 \* Crea la pathFactura y los documentos adjuntos para un tipoDeFactura dado.  
 \* @param planta planta para la que se realiza la pathFactura  
 \* @param mesAFacturar el mes que se va a facturar  
 \* @param fechaDePago fecha en la que se realizó el pago del mes  
 \* @return la ubicación del archivo zip generado con todos los documentos  
 \*/  
 public String facturar(Globales.Plantas planta, YearMonth mesAFacturar,  
 Date fechaDePago) {  
 //Obtiene los datos de la calculadora y los pasa al generador de documentos correspondientes según el tipo de pathFactura  
 System.out.printf("Facturando planta %d, mes: %s, fecha de pago %s...\n", planta.getID().intValue(),  
 mesAFacturar.toString(), fechaDePago.toString());  
 GeneradorDeDocumentosDeFacturacion generadorDeDocumentos = tipoDeFactura.generadorDeDocumentos;  
 DatosDeFacturacion datosDeFacturacion =  
 tipoDeFactura.calculadora.getDatosParaFacturacion(planta, mesAFacturar, fechaDePago,variableManuales);  
 for (Map.Entry<DatosDeFacturacion.Variables, ArrayList<Object>> variable : datosDeFacturacion.valores.entrySet()) {  
 System.out.println("Variable:"+variable.getKey());  
 System.out.println("Valor :"+variable.getValue());  
  
 }  
 setMonto(buscarMontoEnDatos(datosDeFacturacion));  
 generadorDeDocumentos.generarFactura(planta, datosDeFacturacion, mesAFacturar);  
 generadorDeDocumentos.generarDocumentosAdjuntos(planta, datosDeFacturacion, mesAFacturar);  
 return generadorDeDocumentos.generdoarZip(tipoDeFactura);  
 }  
  
 /\*\*  
 \* Convierte un String en un tipo de factura. Si no está definido devuelve INVALIDO.  
 \* @param tipoDeFactura nombre del tipo de factura  
 \* @return tipo de factura {@link TipoDeFactura TipoDeFactura}, con calculadora y generador de documentos definidos  
 \*/  
 public static TipoDeFactura getTipoDeFactura(String tipoDeFactura) {  
 for (TipoDeFactura tipo : TipoDeFactura.values()) {  
 if (tipo.getNombre().equals(tipoDeFactura)) {  
 return tipo;  
 }  
 }  
 return TipoDeFactura.INVALIDO;  
 }  
  
 public void setTipoDeFactura(Facturador.TipoDeFactura tipoDeFactura) {  
 this.tipoDeFactura = tipoDeFactura;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\generadordedocumentos\GeneradorDeDocumentosAOMXLT.java

package mx.mbi.dgc.business.facturacion.facturador.generadordedocumentos;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.global.Globales;  
  
import java.time.YearMonth;  
  
public class GeneradorDeDocumentosAOMXLT extends GeneradorDeDocumentosDeFacturacion {  
  
 public GeneradorDeDocumentosAOMXLT() {  
 super();  
 }  
  
 @Override  
 public void generarDocumentosAdjuntos(Globales.Plantas planta, DatosDeFacturacion datosDeFacturacion, YearMonth mes) {  
 documentos.add(generarDocumento(TiposDeDocumento.F001, planta, datosDeFacturacion, mes));  
 documentos.add(generarDocumento(TiposDeDocumento.F002, planta, datosDeFacturacion, mes));  
 }  
  
 @Override  
 String getCarpetaPEAT(){  
 return "PEAT-006/";  
 }  
  
 @Override  
 String getCarpetaPEST(){  
 return "PEST-006/";  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\generadordedocumentos\GeneradorDeDocumentosDeFacturacion.java

package mx.mbi.dgc.business.facturacion.facturador.generadordedocumentos;  
  
import java.io.File;  
import java.io.IOException;  
  
import java.time.YearMonth;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.Map;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.business.facturacion.facturador.Facturador;  
import mx.mbi.dgc.global.Globales;  
import mx.mbi.dgc.global.Globales.Plantas;  
import mx.mbi.dgc.utils.exportadorexcel.GeneradorExcel;  
import mx.mbi.dgc.utils.zipeador.Zipeador;  
  
public abstract class GeneradorDeDocumentosDeFacturacion {  
 public static String[] corchetes = {".${","}$."};  
  
 /\* Interfaz \*/  
 public abstract void generarDocumentosAdjuntos(Plantas planta, DatosDeFacturacion datosDeFacturacion, YearMonth mes);  
 abstract String getCarpetaPEAT();  
 abstract String getCarpetaPEST();  
  
 /\* Herencia \*/  
 //Tipos de documento y su nombre de archivo correspondiente  
 enum TiposDeDocumento {  
 FACTURA("Factura"),  
 F001("F001"),  
 F002("F002"),  
 F003("F003");  
  
 private final String nombreDeArchivoDePlantilla;  
  
 TiposDeDocumento(String nombreDeArchivo) {  
 nombreDeArchivoDePlantilla = nombreDeArchivo;  
 }  
  
 public String getNombreDePlantilla() {  
 return nombreDeArchivoDePlantilla;  
 }  
 }  
  
 //---Propiedades  
 private static final String tipoDeArchivo = ".xlsx";  
 private static final String carpetaDeSol = "EST/";  
 private static final String carpetaDeAguila = "EAT/";  
 ArrayList<String> documentos;  
 private YearMonth mes;  
 private Plantas planta;  
 private GeneradorExcel generadorExcel;  
  
 //---Constructores  
 GeneradorDeDocumentosDeFacturacion(){  
 documentos = new ArrayList<>();  
 }  
  
 //---Métodos públicos  
  
 /\*\*  
 \* genera la factura  
 \* @param planta planta para la cual se realiza la factura  
 \* @param datosDeFacturacion valores para realizar la factura  
 \* @param mes mes para el cual se realiza la factura  
 \*/  
 public void generarFactura(Plantas planta, DatosDeFacturacion datosDeFacturacion, YearMonth mes){  
 System.out.println("Generando factura");  
 documentos.add(generarDocumento(TiposDeDocumento.FACTURA, planta, datosDeFacturacion, mes));  
 }  
  
 /\*\*  
 \* Genera un documento de un tipo dado, para una planta dada.  
 \* @param tipoDeDocumento el tipo de documento a generar  
 \* @param planta la planta para la cual se genera el documento  
 \* @param datosDeFacturacion contiene los datos y valores para generar el documento  
 \* @return la ubicación del documento generado  
 \*/  
 String generarDocumento(TiposDeDocumento tipoDeDocumento,  
 Plantas planta, DatosDeFacturacion datosDeFacturacion, YearMonth mes) {  
 this.mes = mes;  
 this.planta = planta;  
  
 System.out.printf("Generando documento %s para %s\n", tipoDeDocumento.getNombreDePlantilla(), mes.toString());  
 //Recupera la ubicación de la plantilla del documento según el tipo de factura y su fecha  
 String ubicacionDePlantilla = getPath(Globales.PATH\_FACTURACION\_PLANTILLAS);  
 String ubicacionDeDocumento = getPath(Globales.PATH\_FACTURACION\_DOCUMENTOS);  
  
 String sMes = mes.toString();  
 ubicacionDeDocumento += sMes + "/";  
  
 crearDirectorio(ubicacionDeDocumento);  
  
 //Tienen el mismo nombre pero una ubicación diferente.  
 ubicacionDePlantilla += tipoDeDocumento.getNombreDePlantilla() + tipoDeArchivo;  
 ubicacionDeDocumento += tipoDeDocumento.getNombreDePlantilla() + tipoDeArchivo;  
  
 crearExcel(datosDeFacturacion, ubicacionDePlantilla, ubicacionDeDocumento);  
  
 return ubicacionDeDocumento; //Retorna el path del documento generado y rellenado.  
  
 }  
  
 /\*\*  
 \* Genera un archivo .zip  
 \* @param tipoDeFactura tipo de factura a exportar  
 \* @return la ubicación del zip  
 \*/  
 public String generarZip(Facturador.TipoDeFactura tipoDeFactura) {  
 System.out.printf("Generando ZIP para %s, %s, %s\n", planta, tipoDeFactura.getNombre(), mes.toString());  
 String ubicacionZip = getPath(Globales.PATH\_FACTURACION\_ZIPS);  
  
 crearDirectorio(ubicacionZip);  
  
 ubicacionZip += tipoDeFactura.getNombre() + ".zip"; //¿getNombre es redundante?  
  
 System.out.printf("Creando zip en %s\n", ubicacionZip);  
 Zipeador.generarZip(ubicacionZip, documentos);  
  
 return ubicacionZip;  
 }  
  
 //---Métodos privados  
  
 private void crearExcel(DatosDeFacturacion datosDeFacturacion, String ubicacionDePlantilla, String ubicacionDeDocumento) {  
 //Lee y almacena las ubicaciones de los placeHolders para las variables  
 try  
 {  
 generadorExcel = new GeneradorExcel(ubicacionDePlantilla);  
 HashMap<String, ArrayList<Integer[]>> placeHolders = getPlaceHoldersDelArchivo(generadorExcel);  
  
// if (placeHolders == null) {  
// System.out.printf("No se detectaron placeholders definidos para %s\n", tipoDeDocumento.getNombreDePlantilla());  
// return "";  
// }  
  
 substituirPlaceholdersPorValores(placeHolders, datosDeFacturacion);  
 guardarDocumento(ubicacionDeDocumento);  
 }catch(Exception e) {  
 System.out.println("Error al general excel");  
 e.printStackTrace();  
 }  
 }  
  
 private void guardarDocumento(String ubicacionDeDocumento) {  
 //Exportador Excel guarda el archivo generado y devuelve su ubicación  
 try {  
 generadorExcel.finalizarEscritura(ubicacionDeDocumento);  
 generadorExcel.cerrarLector();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 private void substituirPlaceholdersPorValores(HashMap<String, ArrayList<Integer[]>> placeHolders,  
 DatosDeFacturacion datosDeFacturacion) {  
 //TODO: substituir placeholders para listas de valores.  
 //Por cada valor definido para los datosDeFacturacion de la factura  
 for (Map.Entry<DatosDeFacturacion.Variables, ArrayList<Object>> variable : datosDeFacturacion.valores.entrySet()) {  
 ArrayList<Integer[]> ubicaciones = placeHolders.get(variable.getKey().getSimbolo());  
 if (ubicaciones == null) {  
 System.out.printf("No se encuentra place holder para %s\n",variable.getKey().getNombre());  
 } else {  
 for (Integer[] ubicacion: ubicaciones) {  
 generadorExcel.actualizarCelda(ubicacion, variable.getValue().get(variable.getValue().size()-1));  
 }  
 }  
  
 }  
 }  
  
 private HashMap<String, ArrayList<Integer[]>> getPlaceHoldersDelArchivo(GeneradorExcel generadorExcel) {  
 HashMap<String, ArrayList<Integer[]>> placeHolders = generadorExcel.escanearExcel(corchetes[0],corchetes[1]);  
  
 System.out.println("PlaceHolders encontrados:");  
 for (String placeHolder: placeHolders.keySet()) {  
 System.out.println(placeHolder);  
 }  
 return placeHolders;  
 }  
  
 private void crearDirectorio(String directorio){  
 boolean existe = !new File(directorio).mkdirs();  
 if (existe){  
 System.out.printf("El directorio <<%s>> ya existe\n", directorio);  
 }else {  
 System.out.printf("El directorio <<%s>> fue creado\n", directorio);  
 }  
 }  
  
 private String getPath(String directorioPorTipoDeArchivo) {  
 String ubicacion = directorioPorTipoDeArchivo;  
 if (planta == Plantas.Aguila) {  
 ubicacion += carpetaDeAguila + getCarpetaPEAT();  
 } else {  
 ubicacion += carpetaDeSol + getCarpetaPEST();  
 }  
  
 return ubicacion;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\generadordedocumentos\GeneradorDeDocumentosINVALIDO.java

package mx.mbi.dgc.business.facturacion.facturador.generadordedocumentos;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.business.facturacion.facturador.Facturador;  
import mx.mbi.dgc.global.Globales;  
  
import java.time.YearMonth;  
  
/\*\*  
 \* Cuando se elige un tipo de documento inválido, no se generan los documentos y en su lugar se dan mensajes de error  
 \*/  
public class GeneradorDeDocumentosINVALIDO extends GeneradorDeDocumentosDeFacturacion {  
 @Override  
 String getCarpetaPEAT() {  
 System.out.println("No existe carpeta PEAT para tipo de factura inválido");  
 return "";  
 }  
  
 @Override  
 String getCarpetaPEST() {  
 System.out.println("No existe carpeta PEST para tipo de factura inválido");  
 return "";  
 }  
  
 @Override  
 String generarDocumento(TiposDeDocumento tipoDeDocumento, Globales.Plantas planta, DatosDeFacturacion datosDeFacturacion, YearMonth mes){  
 System.out.println("Tipo de Factura invalido");  
 return null;  
 }  
  
 @Override  
 public void generarDocumentosAdjuntos(Globales.Plantas planta, DatosDeFacturacion datosDeFacturacion, YearMonth mes){  
 System.out.println("No se pueden generar Documentos Adjuntos");  
 }  
  
 @Override  
 public void generarFactura(Globales.Plantas planta, DatosDeFacturacion datosDeFacturacion, YearMonth mes){  
 System.out.println("No se puede generar factura para tipo de factura inválido");  
 }  
  
 @Override  
 public String generarZip(Facturador.TipoDeFactura tipoDeFactura){  
 System.out.println("No se puede generar factura para tipo de factura inválido");  
 return "";  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\generadordedocumentos\GeneradorDeDocumentosPCA.java

package mx.mbi.dgc.business.facturacion.facturador.generadordedocumentos;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.global.Globales;  
  
import java.time.YearMonth;  
  
public class GeneradorDeDocumentosPCA extends GeneradorDeDocumentosDeFacturacion {  
 @Override  
 public void generarDocumentosAdjuntos(Globales.Plantas planta, DatosDeFacturacion datosDeFacturacion, YearMonth mes) {  
  
 }  
  
 @Override  
 public String getCarpetaPEAT() {  
 return "PEAT-005/";  
 }  
  
 @Override  
 public String getCarpetaPEST() {  
 return "PEST-005/";  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\generadordedocumentos\GeneradorDeDocumentosPCC.java

package mx.mbi.dgc.business.facturacion.facturador.generadordedocumentos;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.global.Globales;  
  
import java.time.YearMonth;  
  
public class GeneradorDeDocumentosPCC extends GeneradorDeDocumentosDeFacturacion {  
 public GeneradorDeDocumentosPCC() {  
 super();  
 }  
  
 @Override  
 public void generarDocumentosAdjuntos(Globales.Plantas planta, DatosDeFacturacion datosDeFacturacion, YearMonth mes) {  
 documentos.add(generarDocumento(TiposDeDocumento.F001, planta, datosDeFacturacion, mes));  
 documentos.add(generarDocumento(TiposDeDocumento.F002, planta, datosDeFacturacion, mes));  
 }  
  
 @Override  
 public String getCarpetaPEAT() {  
 return "PEAT-004/";  
 }  
  
 @Override  
 public String getCarpetaPEST() {  
 return "PEST-004/";  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\generadordedocumentos\GeneradorDeDocumentosPCFC.java

package mx.mbi.dgc.business.facturacion.facturador.generadordedocumentos;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.global.Globales.Plantas;  
  
import java.time.YearMonth;  
  
  
/\*\*  
 \* Extiende GeneradorDeDocumentosDeFacturacion. Define los documentos adjuntos y las carpetas de PEAT y PEST.  
 \*/  
public class GeneradorDeDocumentosPCFC extends GeneradorDeDocumentosDeFacturacion {  
 //Se recuperan por medio de getters para que así se reflejen en la clase abstracta padre  
 private static final String carpetaPEST = "PEST-001/";  
 private static final String carpetaPEAT = "PEAT-001/";  
  
 public GeneradorDeDocumentosPCFC() {  
 super();  
 }  
  
 @SuppressWarnings("Duplicates")  
 @Override  
 public void generarDocumentosAdjuntos(Plantas planta, DatosDeFacturacion datosDeFacturacion, YearMonth mes) {  
 documentos.add(generarDocumento(TiposDeDocumento.F001, planta, datosDeFacturacion, mes));  
 documentos.add(generarDocumento(TiposDeDocumento.F002, planta, datosDeFacturacion, mes));  
 documentos.add(generarDocumento(TiposDeDocumento.F003, planta, datosDeFacturacion, mes));  
 }  
  
 @Override  
 String getCarpetaPEAT(){  
 return carpetaPEAT;  
 }  
  
 @Override  
 String getCarpetaPEST(){  
 return carpetaPEST;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\generadordedocumentos\GeneradorDeDocumentosPCFOM.java

package mx.mbi.dgc.business.facturacion.facturador.generadordedocumentos;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.global.Globales.Plantas;  
  
import java.time.YearMonth;  
  
  
/\*\*  
 \* Extiende GeneradorDeDocumentosDeFacturacion. Define los documentos adjuntos y las carpetas de PEAT y PEST.  
 \*/  
public class GeneradorDeDocumentosPCFOM extends GeneradorDeDocumentosDeFacturacion {  
 //Se recuperan por medio de getters para que así se reflejen en la clase abstracta padre  
 private static final String carpetaPEST = "PEST-002/";  
 private static final String carpetaPEAT = "PEAT-002/";  
  
 public GeneradorDeDocumentosPCFOM() {  
 super();  
 }  
  
 @SuppressWarnings("Duplicates")  
 @Override  
 public void generarDocumentosAdjuntos(Plantas planta, DatosDeFacturacion datosDeFacturacion, YearMonth mes) {  
 documentos.add(generarDocumento(TiposDeDocumento.F001, planta, datosDeFacturacion, mes));  
 documentos.add(generarDocumento(TiposDeDocumento.F002, planta, datosDeFacturacion, mes));  
 documentos.add(generarDocumento(TiposDeDocumento.F003, planta, datosDeFacturacion, mes));  
 }  
  
 @Override  
 String getCarpetaPEAT(){  
 return carpetaPEAT;  
 }  
  
 @Override  
 String getCarpetaPEST(){  
 return carpetaPEST;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\facturador\generadordedocumentos\GeneradorDeDocumentosPCVOM.java

package mx.mbi.dgc.business.facturacion.facturador.generadordedocumentos;  
  
import java.time.YearMonth;  
  
import mx.mbi.dgc.business.facturacion.facturador.datosdefacturacion.DatosDeFacturacion;  
import mx.mbi.dgc.global.Globales.Plantas;  
  
public class GeneradorDeDocumentosPCVOM extends GeneradorDeDocumentosDeFacturacion {  
 static final String carpetaPEST = "PEST-003/";  
 static final String carpetaPEAT = "PEAT-003/";  
  
 public GeneradorDeDocumentosPCVOM() {  
 super();  
 }  
  
 @SuppressWarnings("Duplicates")  
 @Override  
 public void generarDocumentosAdjuntos(Plantas planta, DatosDeFacturacion datosDeFacturacion, YearMonth mes) {  
 documentos.add(generarDocumento(TiposDeDocumento.F001, planta, datosDeFacturacion, mes));  
 documentos.add(generarDocumento(TiposDeDocumento.F002, planta, datosDeFacturacion, mes));  
 documentos.add(generarDocumento(TiposDeDocumento.F003, planta, datosDeFacturacion, mes));  
 }  
  
 @Override  
 String getCarpetaPEAT(){  
 return carpetaPEAT;  
 }  
  
 @Override  
 String getCarpetaPEST(){  
 return carpetaPEST;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\IfaceBusinessFacturacion.java

package mx.mbi.dgc.business.facturacion;  
  
import java.util.List;  
  
import mx.mbi.dgc.dominio.DominioCatCompaniaPlantas;  
import mx.mbi.dgc.dominio.DominioCatPlantaTipoFacturas;  
import mx.mbi.dgc.dominio.DominioCatServicioImpuestos;  
  
public interface IfaceBusinessFacturacion {  
 int getMaxFolioBySerie(Object obj) throws Exception;  
  
 DominioCatCompaniaPlantas findCompaniaPlantaByPlantaId(Object obj) throws Exception;  
  
 List<DominioCatServicioImpuestos> findImpuestosByServicio(Object obj) throws Exception;  
  
 List<DominioCatPlantaTipoFacturas> findTipoFacturasByCatPlanta(Object obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\IfaceFacturacionSelladoTimbrado.java

package mx.mbi.dgc.business.facturacion;  
  
import mx.mbi.dgc.dominio.DominioTxDocumentos;  
  
  
public interface IfaceFacturacionSelladoTimbrado {  
  
 public String cfdiCadenaOriginal(DominioTxDocumentos txDocumentos);  
  
 public String creaCertificado(String file\_certificado);  
  
 public String creaSello(String cadenaOriginal, String keyFile, String password);  
  
 public String timbra(byte[] str\_xml);  
  
 public String cfdiXML(DominioTxDocumentos cfdi);  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\facturacion\IfaceMontosFactura.java

package mx.mbi.dgc.business.facturacion;  
  
import mx.mbi.dgc.dominio.negocio.facturacion.calculos.DominioCalculosCargosFijos;  
  
public interface IfaceMontosFactura {  
  
 public DominioCalculosCargosFijos montosPCFC(DominioCalculosCargosFijos dom);  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatAnioMes.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatAnioMes;  
  
public interface IfaceCatAnioMes extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatAnioMes obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatCapacidad.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCapacidad;  
  
public interface IfaceCatCapacidad extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatCapacidad obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatCargoFijo.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCargoFijo;  
  
public interface IfaceCatCargoFijo extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatCargoFijo obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatCargos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCargos;  
  
public interface IfaceCatCargos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatCargos obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatClasEventos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatClasificacionEventos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
public interface IfaceCatClasEventos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatClasificacionEventos obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatClasificacion.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatClasificacion;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
public interface IfaceCatClasificacion extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatClasificacion obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatClasificacionEvento.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatClasificacionEvento;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
public interface IfaceCatClasificacionEvento extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatClasificacionEvento obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatClasificacionEventos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatClasificacionEventos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
public interface IfaceCatClasificacionEventos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatClasificacionEventos obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatClienteMonedas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatClienteMonedas;  
  
public interface IfaceCatClienteMonedas extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatClienteMonedas obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatClientes.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatClientes;  
  
public interface IfaceCatClientes extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatClientes obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatCombustible.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCombustible;  
  
public interface IfaceCatCombustible extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatCombustible obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatCompania.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCompania;  
  
public interface IfaceCatCompania extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatCompania obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatCompaniaPlantas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCompaniaPlantas;  
  
public interface IfaceCatCompaniaPlantas extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatCompaniaPlantas obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatComponentePeriodo.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatComponentePeriodo;  
  
public interface IfaceCatComponentePeriodo extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatComponentePeriodo obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatComponentes.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatComponentes;  
  
public interface IfaceCatComponentes extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatComponentes obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatCondiciones.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCondiciones;  
  
public interface IfaceCatCondiciones extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatCondiciones obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatConexiones.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatConexiones;  
  
public interface IfaceCatConexiones extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatConexiones obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatConfiguracionBitacora.java

package mx.mbi.dgc.business;  
  
import java.util.List;  
  
import mx.mbi.dgc.dominio.DominioCatDetalleOpciones;  
import mx.mbi.dgc.dominio.DominioCatDetalles;  
import mx.mbi.dgc.dominio.DominioCatOpcionPlanta;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
public interface IfaceCatConfiguracionBitacora extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatDetalleOpciones obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatConfiguracionDetalle.java

package mx.mbi.dgc.business;  
  
import java.util.List;  
  
import mx.mbi.dgc.dominio.DominioCatConfiguracionDetalle;  
import mx.mbi.dgc.dominio.DominioCatConfiguracionEventos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
public interface IfaceCatConfiguracionDetalle extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatConfiguracionDetalle obj) throws Exception;  
  
 List<DominioCatConfiguracionDetalle> findByCatConfiguracionEvento(DominioCatConfiguracionEventos dom) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatConfiguracionEventos.java

package mx.mbi.dgc.business;  
  
import java.util.List;  
  
import mx.mbi.dgc.dominio.DominioCatClasificacionEvento;  
import mx.mbi.dgc.dominio.DominioCatConfiguracionEventos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
public interface IfaceCatConfiguracionEventos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatConfiguracionEventos obj) throws Exception;  
  
 List<DominioCatConfiguracionEventos> findByCatClasificacionEvento(DominioCatClasificacionEvento dom) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatConfiguracionOpciones.java

package mx.mbi.dgc.business;  
  
import java.util.List;  
  
import mx.mbi.dgc.dominio.DominioCatConfiguracionOpciones;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.negocio.bitacora.DominioPlantaOpcionesDetalle;  
  
public interface IfaceCatConfiguracionOpciones extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatConfiguracionOpciones obj) throws Exception;  
  
 List<DominioCatConfiguracionOpciones> findByCatConfiguracionDetalle(DominioPlantaOpcionesDetalle dom) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatContratosAfectados.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatContratosAfectados;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
public interface IfaceCatContratosAfectados extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatContratosAfectados obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatCurvasCorreccion.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCurvasCorreccion;  
  
public interface IfaceCatCurvasCorreccion extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatCurvasCorreccion obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatDatosFiscales.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatDatosFiscales;  
  
public interface IfaceCatDatosFiscales extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatDatosFiscales obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatDetalleOpciones.java

package mx.mbi.dgc.business;  
  
import java.util.List;  
  
import mx.mbi.dgc.dominio.DominioCatDetalleOpciones;  
import mx.mbi.dgc.dominio.DominioCatDetalles;  
import mx.mbi.dgc.dominio.DominioCatOpcionPlanta;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
public interface IfaceCatDetalleOpciones extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatDetalleOpciones obj) throws Exception;  
 List<DominioCatDetalleOpciones> findByDetalleId(DominioCatDetalles dominioCatDetalles)throws Exception;  
 List<DominioCatDetalleOpciones> findByDetalle(DominioCatDetalles dominioCatDetalles)throws Exception;  
 List<DominioCatDetalleOpciones> findByDetallePlantaId (DominioCatOpcionPlanta dominioCatOpcionesPlanta) throws Exception;  
 List<DominioCatDetalleOpciones> findCheck (DominioCatOpcionPlanta dominioCatOpcionPlanta) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatDetalles.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatDetalles;  
  
public interface IfaceCatDetalles extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatDetalles obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatEstados.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatEstados;  
  
public interface IfaceCatEstados extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatEstados obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatEstatusBitacora.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatEstatusBitacora;  
  
public interface IfaceCatEstatusBitacora extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatEstatusBitacora obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatEstatusFactura.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatEstatusFactura;  
  
public interface IfaceCatEstatusFactura extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatEstatusFactura obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatEstatusPagos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatEstatusPagos;  
  
public interface IfaceCatEstatusPagos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatEstatusPagos obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatEstatusPerfil.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatEstatusPerfil;  
  
public interface IfaceCatEstatusPerfil extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatEstatusPerfil obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatEstatusVariable.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatEstatusVariable;  
  
public interface IfaceCatEstatusVariable extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatEstatusVariable obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatEtapas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatEtapas;  
  
public interface IfaceCatEtapas extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatEtapas obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatEventos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatEventos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
public interface IfaceCatEventos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatEventos obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatFactorTipo.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatFactorTipo;  
  
public interface IfaceCatFactorTipo extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatFactorTipo obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatFacturaProductos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatFacturaProductos;  
  
public interface IfaceCatFacturaProductos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatFacturaProductos obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatFirmaCincominutal.java

package mx.mbi.dgc.business;  
  
import java.util.List;  
  
import mx.mbi.dgc.dominio.DominioCatFirmaCincominutal;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
public interface IfaceCatFirmaCincominutal extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatFirmaCincominutal obj) throws Exception;  
 List<DominioCatFirmaCincominutal> findAllLimit() throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatFormasPago.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatFormasPago;  
  
public interface IfaceCatFormasPago extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatFormasPago obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatFormulaComponentes.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatFormulaComponentes;  
  
public interface IfaceCatFormulaComponentes extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatFormulaComponentes obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatFormulaComponentesTxEvento.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatFormulaComponentesTxEvento;  
  
public interface IfaceCatFormulaComponentesTxEvento extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatFormulaComponentesTxEvento obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatImpuestos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatImpuestos;  
  
public interface IfaceCatImpuestos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatImpuestos obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatLimitesDominio.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatLimitesDominio;  
  
public interface IfaceCatLimitesDominio extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatLimitesDominio obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatMedidor.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatMedidor;  
  
public interface IfaceCatMedidor extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatMedidor obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatMedidores.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatMedidores;  
  
public interface IfaceCatMedidores extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatMedidores obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatMesCargo.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatMesCargo;  
  
public interface IfaceCatMesCargo extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatMesCargo obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatMesDia.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatMesDia;  
  
public interface IfaceCatMesDia extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatMesDia obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatMesDiaFirma.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatMesDiaFirma;  
  
public interface IfaceCatMesDiaFirma extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatMesDiaFirma obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatMetodoPago.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatMetodoPago;  
  
public interface IfaceCatMetodoPago extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatMetodoPago obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatMonedas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatMonedas;  
  
public interface IfaceCatMonedas extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatMonedas obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatOpcionPlanta.java

package mx.mbi.dgc.business;  
  
import java.util.List;  
  
import mx.mbi.dgc.dominio.DominioCatDetalleOpciones;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatOpcionPlanta;  
  
public interface IfaceCatOpcionPlanta extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatOpcionPlanta obj) throws Exception;  
 DominioCatOpcionPlanta findByOpcionPlanta (DominioCatOpcionPlanta dominioCatOpcionPlanta) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatPaises.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatPaises;  
  
public interface IfaceCatPaises extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatPaises obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatPermiso.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatPermiso;  
  
public interface IfaceCatPermiso extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatPermiso obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatPlanta.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatPlanta;  
  
public interface IfaceCatPlanta extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatPlanta obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatPlantaTipoFacturas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatPlantaTipoFacturas;  
  
public interface IfaceCatPlantaTipoFacturas extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatPlantaTipoFacturas obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatPropCincominutal.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatPropCincominutal;  
  
public interface IfaceCatPropCincominutal extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatPropCincominutal obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatQuery.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatQuery;  
  
public interface IfaceCatQuery extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatQuery obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatRangoVariable.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatRangoVariable;  
  
public interface IfaceCatRangoVariable extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatRangoVariable obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatRegimenFiscal.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatRegimenFiscal;  
  
public interface IfaceCatRegimenFiscal extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatRegimenFiscal obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatReglaNegocio.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatReglaNegocio;  
  
public interface IfaceCatReglaNegocio extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatReglaNegocio obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatRol.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatRol;  
  
public interface IfaceCatRol extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatRol obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatServicioImpuestos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatServicioImpuestos;  
  
public interface IfaceCatServicioImpuestos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatServicioImpuestos obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatServicios.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatServicios;  
  
public interface IfaceCatServicios extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatServicios obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatSistemas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatSistemas ;  
  
public interface IfaceCatSistemas extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatSistemas obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatSubCargos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatSubCargos;  
  
public interface IfaceCatSubCargos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatSubCargos obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatTags.java

package mx.mbi.dgc.business;  
  
import java.util.List;  
  
import mx.mbi.dgc.dominio.DominioCatPlanta;  
import mx.mbi.dgc.dominio.DominioCatTags;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
public interface IfaceCatTags extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatTags obj) throws Exception;  
  
 List<DominioCatTags> findByPlantaId(DominioCatPlanta dom) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatTipoDato.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioCatTipoDato;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
public interface IfaceCatTipoDato extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatTipoDato obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatTipoDocumento.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatTipoDocumento;  
  
public interface IfaceCatTipoDocumento extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatTipoDocumento obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatTipoFactoresSat.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatTipoFactoresSat;  
  
public interface IfaceCatTipoFactoresSat extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatTipoFactoresSat obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatTipoImpuestos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatTipoImpuestos;  
  
public interface IfaceCatTipoImpuestos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatTipoImpuestos obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatTipoRelacion.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatTipoRelacion;  
  
public interface IfaceCatTipoRelacion extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatTipoRelacion obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatTiposDeEvento.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatTiposDeEvento;  
  
public interface IfaceCatTiposDeEvento extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatTiposDeEvento obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatTiposFacturas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatTiposFacturas;  
  
public interface IfaceCatTiposFacturas extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatTiposFacturas obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatTiposServicios.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatTiposServicios;  
  
public interface IfaceCatTiposServicios extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatTiposServicios obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatUnidad.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatUnidad;  
  
public interface IfaceCatUnidad extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatUnidad obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatUnidades.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatUnidades ;  
  
public interface IfaceCatUnidades extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatUnidades obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatUsoCfdi.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatUsoCfdi;  
  
public interface IfaceCatUsoCfdi extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatUsoCfdi obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatVariable.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatVariable;  
  
public interface IfaceCatVariable extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatVariable obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatVariablePerfil.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatVariablePerfil;  
  
public interface IfaceCatVariablePerfil extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatVariablePerfil obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceCatVersion.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatVersion;  
  
public interface IfaceCatVersion extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioCatVersion obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceJRGeneric.java

package mx.mbi.dgc.business;  
  
import java.util.List;  
  
public interface IfaceJRGeneric<T> {  
 List<T> findAll() throws Exception;  
 List<T> findAllPagination(T obj) throws Exception;  
 T save(T obj) throws Exception;  
 T update(T obj) throws Exception;  
 T delete(T obj) throws Exception;  
 T findById(T obj) throws Exception;  
 long count() throws Exception;  
 T existsById(T obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceLogin.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioTxUsuario;  
  
public interface IfaceLogin extends IfaceJRGeneric {  
 DominioTxUsuario findByUsername(DominioTxUsuario dom) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceSatImpuestos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatImpuestos;  
  
public interface IfaceSatImpuestos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioSatImpuestos obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceSatMonedas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatMonedas;  
  
public interface IfaceSatMonedas extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioSatMonedas obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceSatPaises.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatPaises;  
  
public interface IfaceSatPaises extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioSatPaises obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceSatProductosClaves.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatProductosClaves;  
  
public interface IfaceSatProductosClaves extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioSatProductosClaves obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceSatProductosUnidades.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatProductosUnidades;  
  
public interface IfaceSatProductosUnidades extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioSatProductosUnidades obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceSatTasaCuota.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatTasaCuota;  
  
public interface IfaceSatTasaCuota extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioSatTasaCuota obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceSatTipoFactor.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatTipoFactor;  
  
public interface IfaceSatTipoFactor extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioSatTipoFactor obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceSatTipoRelacion.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatTipoRelacion;  
  
public interface IfaceSatTipoRelacion extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioSatTipoRelacion obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceSatUsoCfdi.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatUsoCfdi;  
  
public interface IfaceSatUsoCfdi extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioSatUsoCfdi obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxAdjuntos.java

package mx.mbi.dgc.business;  
  
import java.util.List;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxAdjuntos;  
import mx.mbi.dgc.dominio.DominioTxEvento;  
  
public interface IfaceTxAdjuntos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxAdjuntos obj) throws Exception;  
  
 List<DominioTxAdjuntos> findByEventoId(DominioTxEvento evento) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxBitacora.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxBitacora;  
  
public interface IfaceTxBitacora extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxBitacora obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxCargaHistorico.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxCargaHistorico;  
  
public interface IfaceTxCargaHistorico extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxCargaHistorico obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxCincominutal.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxCincominutal;  
  
public interface IfaceTxCincominutal extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxCincominutal obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxCincominutalVar.java

package mx.mbi.dgc.business;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxCincominutalVar;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
  
public interface IfaceTxCincominutalVar extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxCincominutalVar obj) throws Exception;  
  
 List findByEstatusCnNotAndFcPrimera(BigDecimal estatusCn,CatPlanta catPlanta) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxCrudos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxCrudos;  
  
public interface IfaceTxCrudos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxCrudos obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxCrudosLimpios.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxCrudosLimpios;  
  
public interface IfaceTxCrudosLimpios extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxCrudosLimpios obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxCrudosLimpiosSol.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxCrudosLimpiosSol;  
  
public interface IfaceTxCrudosLimpiosSol extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxCrudosLimpiosSol obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxCrudosSol.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxCrudosSol;  
  
public interface IfaceTxCrudosSol extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxCrudosSol obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxDocumentos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxDocumentos;  
  
public interface IfaceTxDocumentos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxDocumentos obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxErrorCargaCrudos.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxErrorCargaCrudos;  
  
public interface IfaceTxErrorCargaCrudos extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxErrorCargaCrudos obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxErrorFormato.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxErrorFormato;  
  
public interface IfaceTxErrorFormato extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxErrorFormato obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxEscenario.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxEscenario;  
  
public interface IfaceTxEscenario extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxEscenario obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxEscenariosBitacora.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxEscenariosBitacora;  
  
public interface IfaceTxEscenariosBitacora extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxEscenariosBitacora obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxEscenarioVariable.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxEscenarioVariable;  
  
public interface IfaceTxEscenarioVariable extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxEscenarioVariable obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxEvento.java

package mx.mbi.dgc.business;  
  
import java.util.List;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxEvento;  
import mx.mbi.dgc.dominio.negocio.bitacora.DominioBusquedaEventos;  
import mx.mbi.dgc.dominio.negocio.bitacora.DominioEventoBitacora;  
  
public interface IfaceTxEvento extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxEvento obj) throws Exception;  
  
 DominioTxEvento guardarEvento(DominioEventoBitacora dom) throws Exception;  
   
 List<DominioTxEvento> findByBitacora(DominioBusquedaEventos dom) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxEventoDetalles.java

package mx.mbi.dgc.business;  
  
import java.util.List;  
  
import mx.mbi.dgc.dominio.DominioTxEvento;  
import mx.mbi.dgc.dominio.DominioTxEventoDetalles;  
  
  
public interface IfaceTxEventoDetalles extends IfaceJRGeneric {  
 List<DominioTxEventoDetalles> findByTxEvento(DominioTxEvento eventoId) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxFacturaUuids.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxFacturaUuids;  
  
public interface IfaceTxFacturaUuids extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxFacturaUuids obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxFormulas.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxFormulas;  
  
public interface IfaceTxFormulas extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxFormulas obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxLogErrorCarga.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxLogErrorCarga;  
  
public interface IfaceTxLogErrorCarga extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxLogErrorCarga obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxPerfilesVariable.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxPerfilesVariable;  
  
public interface IfaceTxPerfilesVariable extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxPerfilesVariable obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxProduccion.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxProduccion;  
  
public interface IfaceTxProduccion extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxProduccion obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxRolPermiso.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxRolPermiso;  
  
public interface IfaceTxRolPermiso extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxRolPermiso obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxTrazabilidad.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxTrazabilidad;  
  
public interface IfaceTxTrazabilidad extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxTrazabilidad obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxTrazaQuery.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxTrazaQuery;  
  
public interface IfaceTxTrazaQuery extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxTrazaQuery obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxUsuario.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxUsuario;  
  
public interface IfaceTxUsuario extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxUsuario obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxUsuarioPlanta.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxUsuarioPlanta;  
  
public interface IfaceTxUsuarioPlanta extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxUsuarioPlanta obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\IfaceTxUsuarioRol.java

package mx.mbi.dgc.business;  
  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxUsuarioRol;  
  
public interface IfaceTxUsuarioRol extends IfaceJRGeneric {  
 DominioRespuesta allPagination(DominioTxUsuarioRol obj) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\mm\BusinessModeloMatematico.java

package mx.mbi.dgc.business.mm;  
  
import java.math.BigDecimal;  
  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import mx.mbi.dgc.client.exe.eat.ArrayOfEATInput;  
import mx.mbi.dgc.client.exe.eat.ArrayOfEATOutput;  
import mx.mbi.dgc.client.exe.eat.EATInput;  
import mx.mbi.dgc.client.exe.eat.EATMModel;  
import mx.mbi.dgc.client.exe.eat.EATOutput;  
import mx.mbi.dgc.client.exe.eat.EATSingle;  
import mx.mbi.dgc.client.exe.est.ArrayOfESTInput;  
import mx.mbi.dgc.client.exe.est.ArrayOfESTOutput;  
import mx.mbi.dgc.client.exe.est.ESTInput;  
import mx.mbi.dgc.client.exe.est.ESTMModel;  
import mx.mbi.dgc.client.exe.est.ESTOutput;  
import mx.mbi.dgc.client.exe.est.ESTSingle;  
import mx.mbi.dgc.dao.DAOTxCincominutal;  
import mx.mbi.dgc.dao.DAOTxCincominutalVar;  
import mx.mbi.dgc.dao.negocio.DAOCincoVar;  
import mx.mbi.dgc.dao.negocio.DaoNegocioTxPerfilesVariable;  
import mx.mbi.dgc.dominio.DominioDyCNorma;  
import mx.mbi.dgc.dominio.negocio.DominioEAT;  
import mx.mbi.dgc.dominio.negocio.DominioEST;  
import mx.mbi.dgc.dominio.negocio.DominioModeloMatematico;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxCincominutal;  
import mx.mbi.dgc.jpa.entity.TxCincominutalVar;  
import mx.mbi.dgc.jpa.entity.TxPerfilesVariable;  
import mx.mbi.dgc.global.Globales;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessModeloMatematico implements IfaceModeloMatematico {  
 private static final Logger LOG = LogManager.getLogger(BusinessModeloMatematico.class);  
 private ESTMModel clientESTService;  
 private EATMModel clientEATService;  
 @Autowired  
 private DAOCincoVar daoCincoVar;  
 @Autowired  
 private DaoNegocioTxPerfilesVariable daoTxPerfilesVariable;  
 @Autowired  
 private DAOTxCincominutalVar datTxCincominutalVar;  
 @Autowired  
 private DAOTxCincominutal daoTxCincominutal;  
  
 public BusinessModeloMatematico() {  
 try {  
 clientESTService = new ESTMModel();  
 clientEATService = new EATMModel();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public DominioModeloMatematico calculaFactores(DominioModeloMatematico dom) throws DyCNormaException {  
 System.out.println("#### BusinessModeloMatematico.calculaFactores: ");  
 System.out.println("dom.getPlanta().getPlantaId().intValue(): " + dom.getPlanta()  
 .getPlantaId()  
 .intValue());  
 try {  
 if (dom.getPlanta()  
 .getPlantaId()  
 .intValue() == Globales.EAT) {  
 System.out.println("invocando el webservice de EAT");  
 dom = callEAT(dom);  
 }  
 if (dom.getPlanta()  
 .getPlantaId()  
 .intValue() == Globales.EST) {  
 System.out.println("invocando el webservice de EST");  
 dom = callEST(dom);  
 }  
 dom.setStatus(true);  
 dom.setCode(200);  
 //System.out.println("######### dom.getDetalleCVC.size(): "+dom.getDetalleCVC().size());  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en BusinessModeloMatematico.calculaFactores: " + ex);  
 }  
 return dom;  
 }  
  
 private DominioModeloMatematico callEAT(DominioModeloMatematico dom) throws DyCNormaException {  
 System.out.println("#### BusinessModeloMatematico.callEAT: ");  
  
 try {  
 System.out.println("dom.getTipoCalculo(): " + dom.getTipoCalculo());  
 if (dom.getTipoCalculo().equals("single")) {  
 EATSingle eatSingle = new EATSingle();  
 EATInput input = new EATInput();  
 DominioEAT domEAT = dom.getCincominutalEAT();  
  
 System.out.println("domEAT.getAtmosphericPressure(): " + domEAT.getAtmosphericPressure());  
 System.out.println("domEAT.getCalorificPower(): " + domEAT.getCalorificPower());  
 System.out.println("domEAT.getDryBulbTemp(): " + domEAT.getDryBulbTemp());  
 System.out.println("domEAT.getFuelType(): " + domEAT.getFuelType());  
 System.out.println("domEAT.getCondenserPressure(): " + domEAT.getCapacitorPressure());  
 System.out.println("domEAT.getCurrentCapacity(): " + domEAT.getCurrentCapacity());  
 System.out.println("domEAT.getHumidityRelative(): " + domEAT.getHumidityRelative());  
 System.out.println("domEAT.getPowerFactor(): " + domEAT.getPowerFactor());  
 System.out.println("domEAT.getManualNetCapacity(): " + domEAT.getManualNetCapacity());  
 System.out.println("domEAT.getProvenNetCapacityGas(): " + domEAT.getProvenNetCapacityGas());  
 System.out.println("domEAT.getProvenNetCapacityOil(): " + domEAT.getProvenNetCapacityOil());  
 System.out.println("domEAT.getAvailableEnergy(): " + domEAT.getAvailableEnergy());  
  
 JRUtil.copyProperties(input, domEAT);  
  
 System.out.println("input.getAtmosphericPressure(): " + input.getAtmosphericPressure());  
 System.out.println("input.getCalorificPower(): " + input.getCalorificPower());  
 System.out.println("input.getDryBulbTemp(): " + input.getDryBulbTemp());  
 System.out.println("input.getFuelType(): " + input.getFuelType());  
 System.out.println("input.getCondenserPressure(): " + input.getCapacitorPressure());  
 System.out.println("input.getCurrentCapacity(): " + input.getCurrentCapacity());  
 System.out.println("input.getHumidityRelative(): " + input.getHumidityRelative());  
 System.out.println("input.getPowerFactor(): " + input.getPowerFactor());  
 System.out.println("input.getManualNetCapacity(): " + input.getManualNetCapacity());  
 System.out.println("input.getProvenNetCapacityGas(): " + input.getProvenNetCapacityGas());  
 System.out.println("input.getProvenNetCapacityOil(): " + input.getProvenNetCapacityOil());  
 System.out.println("input.getAvailableEnergy(): " + input.getAvailableEnergy());  
  
 eatSingle.setInput(input);  
 EATOutput response = clientEATService.getEATMModelSoap12().eatSingle(input);  
 System.out.println("respondí MM");  
 JRUtil.copyProperties(domEAT, response);  
 domEAT.setCtungLoad(response.getCTUNGLoad());  
 domEAT.setCtung(response.getCTUNG());  
 dom.setCincominutalEAT(domEAT);  
 }  
 if (dom.getTipoCalculo().equals("batch")) {  
 ArrayOfEATInput batch = new ArrayOfEATInput();  
 List<DominioEAT> lista = dom.getListaEAT();  
  
 for (DominioEAT obj : lista) {  
 EATInput input = new EATInput();  
  
 System.out.println("domEAT.getAtmosphericPressure(): " + obj.getAtmosphericPressure());  
 System.out.println("domEAT.getCalorificPower(): " + obj.getCalorificPower());  
 System.out.println("domEAT.getDryBulbTemp(): " + obj.getDryBulbTemp());  
 System.out.println("domEAT.getFuelType(): " + obj.getFuelType());  
 System.out.println("domEAT.getCondenserPressure(): " + obj.getCapacitorPressure());  
 System.out.println("domEAT.getCurrentCapacity(): " + obj.getCurrentCapacity());  
 System.out.println("domEAT.getHumidityRelative(): " + obj.getHumidityRelative());  
 System.out.println("domEAT.getPowerFactor(): " + obj.getPowerFactor());  
 System.out.println("domEAT.getManualNetCapacity(): " + obj.getManualNetCapacity());  
 System.out.println("domEAT.getProvenNetCapacityGas(): " + obj.getProvenNetCapacityGas());  
 System.out.println("domEAT.getProvenNetCapacityOil(): " + obj.getProvenNetCapacityOil());  
 System.out.println("domEAT.getAvailableEnergy(): " + obj.getAvailableEnergy());  
  
 JRUtil.copyProperties(input, obj);  
 System.out.println("input.getAtmosphericPressure(): " + input.getAtmosphericPressure());  
 System.out.println("input.getCalorificPower(): " + input.getCalorificPower());  
 System.out.println("input.getDryBulbTemp(): " + input.getDryBulbTemp());  
 System.out.println("input.getFuelType(): " + input.getFuelType());  
 System.out.println("input.getCondenserPressure(): " + input.getCapacitorPressure());  
 System.out.println("input.getCurrentCapacity(): " + input.getCurrentCapacity());  
 System.out.println("input.getHumidityRelative(): " + input.getHumidityRelative());  
 System.out.println("input.getPowerFactor(): " + input.getPowerFactor());  
 System.out.println("input.getManualNetCapacity(): " + input.getManualNetCapacity());  
 System.out.println("input.getProvenNetCapacityGas(): " + input.getProvenNetCapacityGas());  
 System.out.println("input.getProvenNetCapacityOil(): " + input.getProvenNetCapacityOil());  
 System.out.println("input.getAvailableEnergy(): " + input.getAvailableEnergy());  
 batch.getEATInput().add(input);  
  
 }  
 System.out.println("##### batch.getEATInput() lista size(): " + batch.getEATInput().size());  
 ArrayOfEATOutput response = clientEATService.getEATMModelSoap12().eatBatch(batch);  
 System.out.println("respondió servicio de batch EAT");  
 List<EATOutput> salida = response.getEATOutput();  
 List<DominioEAT> domsSalida = new ArrayList<>();  
 for (EATOutput out : salida) {  
 DominioEAT domEAT = new DominioEAT();  
 JRUtil.copyProperties(domEAT, out);  
 domEAT.setCtungLoad(out.getCTUNGLoad());  
 domEAT.setCtung(out.getCTUNG());  
 domsSalida.add(domEAT);  
 }  
 dom.setListaEAT(domsSalida);  
 }  
 dom.setStatus(true);  
 dom.setCode(200);  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en BusinessModeloMatematico.callEAT: " + ex);  
 }  
 return dom;  
 }  
  
 private DominioModeloMatematico callEST(DominioModeloMatematico dom) throws DyCNormaException {  
 System.out.println("#### BusinessModeloMatematico.callEST: ");  
 try {  
 if (dom.getTipoCalculo().equals("single")) {  
 ESTSingle estSingle = new ESTSingle();  
 ESTInput input = new ESTInput();  
 DominioEST domEST = dom.getCincominutalEST();  
 JRUtil.copyProperties(input, domEST);  
 estSingle.setInput(input);  
 ESTOutput response = clientESTService.getESTMModelSoap12().estSingle(input);  
 JRUtil.copyProperties(domEST, response);  
 domEST.setCtungLoad(response.getCTUNGLoad());  
 domEST.setCtung(response.getCTUNG());  
 domEST.setCtov(response.getCTOV());  
 dom.setCincominutalEST(domEST);  
 }  
 if (dom.getTipoCalculo().equals("batch")) {  
 ArrayOfESTInput batch = new ArrayOfESTInput();  
 List<DominioEST> lista = dom.getListaEST();  
 for (DominioEST obj : lista) {  
 ESTInput input = new ESTInput();  
 JRUtil.copyProperties(input, obj);  
 batch.getESTInput().add(input);  
 }  
 System.out.println("##### batch.getESTInput() lista size(): " + batch.getESTInput().size());  
 ArrayOfESTOutput response = clientESTService.getESTMModelSoap12().estBatch(batch);  
 System.out.println("respondió servicio de batch EST");  
 List<ESTOutput> salida = response.getESTOutput();  
 List<DominioEST> domsSalida = new ArrayList<>();  
 System.out.println("##### ESTOutput lista size(): " + salida.size());  
 for (ESTOutput out : salida) {  
 DominioEST domEST = new DominioEST();  
 JRUtil.copyProperties(domEST, out);  
 domEST.setCtungLoad(out.getCTUNGLoad());  
 domEST.setCtung(out.getCTUNG());  
 domEST.setCtov(out.getCTOV());  
 domsSalida.add(domEST);  
 }  
 dom.setListaEST(domsSalida);  
 }  
 dom.setStatus(true);  
 dom.setCode(200);  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en BusinessModeloMatematico.callEST: " + ex);  
 }  
 return dom;  
 }  
  
 @Override  
 public void actualizaMM(DominioDyCNorma obj, List<TxCincominutalVar> lista) throws DyCNormaException {  
 try {  
 DominioModeloMatematico dom = new DominioModeloMatematico();  
 dom.setPlanta(obj.getCatPlanta());  
 cincos = new ArrayList<>();  
 if (dom.getPlanta()  
 .getPlantaId()  
 .intValue() == Globales.EAT) {  
 Map<BigDecimal, DominioEAT> mapa = new HashMap<>();  
 for (TxCincominutalVar var : lista) {  
 TxCincominutal cinco = var.getTxCincominutal();  
 if (!existCinco(cinco)) {  
 List<TxCincominutalVar> cincominutalVarByCincominutal;  
 cincominutalVarByCincominutal = datTxCincominutalVar.findByTxCincominutal(cinco);  
 DominioEAT eat = new DominioEAT();  
 if (obj.getTipoEnergia().equals("ES")) { //Estación de supervisión  
 eat.setCurrentCapacity(cinco.getCargaCincominutal());  
  
 eat.setManualNetCapacity(cinco.getCargaCincominutal());  
 }  
 if (obj.getTipoEnergia().equals("PERFILES")) { //Perfiles de energis  
 TxPerfilesVariable perfil = daoTxPerfilesVariable.getPerfilForMM(cinco.getCincominutalId());  
 eat.setCurrentCapacity(perfil.getValor());  
  
 eat.setManualNetCapacity(perfil.getValor());  
 }  
 eat.setAvailableEnergy(cinco.getCatCombustible()  
 .getCombustibleId()  
 .intValue() == 0 ? new BigDecimal(Globales.GAS\_CAPACITY) :  
 new BigDecimal(Globales.OIL\_CAPACITY));  
  
 eat.setPowerFactor(cinco.getFactorDePotencia());  
 eat.setFuelType(cinco.getCatCombustible()  
 .getCombustibleId()  
 .intValue());  
 eat.setProvenNetCapacityGas(new BigDecimal(Globales.GAS\_CAPACITY));  
 eat.setProvenNetCapacityOil(new BigDecimal(Globales.OIL\_CAPACITY));  
 for (TxCincominutalVar txVar : cincominutalVarByCincominutal) {  
 int varId = txVar.getCatVariable()  
 .getVariableId()  
 .intValue();  
 if (varId == 14) {  
 eat.setDryBulbTemp(txVar.getValorNorma());  
 }  
 if (varId == 10) {  
 eat.setAtmosphericPressure(txVar.getValorNorma());  
 }  
 if (varId == 8) {  
 eat.setHumidityRelative(txVar.getValorNorma());  
 }  
 if (varId == 11) {  
 eat.setCalorificPower(txVar.getValorNorma());  
 }  
 if (varId == 9) {  
 eat.setCapacitorPressure(txVar.getValorNorma());  
 }  
 /\*if(varId == 17) {  
 est.setPowerFactor(txVar.getValorNorma());  
 }\*/  
 }  
 // aux.add(est);  
 mapa.put(cinco.getCincominutalId(), eat);  
  
 //Single  
 dom.setTipoCalculo("single");  
 dom.setCincominutalEAT(eat);  
 DominioModeloMatematico response = calculaFactores(dom);  
 DominioEAT eatResponse = response.getCincominutalEAT();  
 List<TxCincominutalVar> listaSaves = new ArrayList<>();  
 System.out.println("CpwAT calculado: " + eatResponse.getCpwAT());  
 System.out.println("variables a actualizar: " + cincominutalVarByCincominutal.size());  
 for (TxCincominutalVar txVar : cincominutalVarByCincominutal) {  
 int varId = txVar.getCatVariable()  
 .getVariableId()  
 .intValue();  
 if (varId == 14) {  
 txVar.setFcPrimera(eatResponse.getCpwAT());  
 }  
 if (varId == 10) {  
 txVar.setFcPrimera(eatResponse.getCpwBP());  
 }  
 if (varId == 8) {  
 txVar.setFcPrimera(eatResponse.getCpwRH());  
 }  
 //??  
 if (varId == 11) {  
 txVar.setFcPrimera(eatResponse.getCpwLHV());  
 }  
 //??  
 if (varId == 9) {  
 txVar.setFcPrimera(eatResponse.getCpwVC());  
 }  
 if (varId == 18) {  
 txVar.setFcPrimera(eatResponse.getCpwPF());  
 }  
 txVar.setFechaPc(new Date());  
 if (obj.getTipoEnergia().equals("ES")) { //Estación de supervisión  
 //cinco.setEnergiaPc(estResponse.getSummerCapacity());//La tabla TX\_CINCOMINUTAL debe tener la propiedad energiaPc  
 //daoTxCincominutal.save(cinco);  
  
  
  
  
  
  
  
 }  
 if (obj.getTipoEnergia()  
 .equals("PERFILES")) { //Perfiles de energis  
 TxPerfilesVariable perfil =  
 daoTxPerfilesVariable.getPerfilForMM(cinco.getCincominutalId());  
 perfil.setEnergiaPc(eatResponse.getSummerCapacity());  
 perfil.setFechaPc(new Date());  
 daoTxPerfilesVariable.save(perfil);  
 }  
 listaSaves.add(txVar);  
 }  
 daoCincoVar.saveAll(listaSaves);  
 }  
 cincos.add(cinco);  
 }  
 }  
 if (dom.getPlanta()  
 .getPlantaId()  
 .intValue() == Globales.EST) {  
 // List<DominioEST> aux = new ArrayList<>();  
 Map<BigDecimal, DominioEST> mapa = new HashMap<>();  
 for (TxCincominutalVar var : lista) {  
 TxCincominutal cinco = var.getTxCincominutal();  
 if (!existCinco(cinco)) {  
 List<TxCincominutalVar> cincominutalVarByCincominutal;  
 cincominutalVarByCincominutal = datTxCincominutalVar.findByTxCincominutal(cinco);  
 DominioEST est = new DominioEST();  
 if (obj.getTipoEnergia().equals("ES")) { //Estación de supervisión  
 est.setCurrentCapacity(cinco.getCargaCincominutal());  
 }  
 if (obj.getTipoEnergia().equals("PERFILES")) { //Perfiles de energis  
 TxPerfilesVariable perfil = daoTxPerfilesVariable.getPerfilForMM(cinco.getCincominutalId());  
 est.setCurrentCapacity(perfil.getValor());  
 }  
 est.setPowerFactor(cinco.getFactorDePotencia());  
  
 System.out.println("variables a preparar: " + cincominutalVarByCincominutal.size());  
 for (TxCincominutalVar txVar : cincominutalVarByCincominutal) {  
 int varId = txVar.getCatVariable()  
 .getVariableId()  
 .intValue();  
 if (varId == 1) {  
 est.setDryBulbTemp(txVar.getValorNorma());  
 }  
 if (varId == 3) {  
 est.setAtmosphericPressure(txVar.getValorNorma());  
 }  
 if (varId == 2) {  
 est.setHumidityRelative(txVar.getValorNorma());  
 }  
 if (varId == 4) {  
 est.setCalorificPower(txVar.getValorNorma());  
 }  
 if (varId == 12) {  
 est.setSeaWaterTemp(txVar.getValorNorma());  
 }  
 /\*  
 if (varId == 17) {  
 est.setPowerFactor(txVar.getValorNorma());  
 }\*/  
 }  
 // aux.add(est);  
 mapa.put(cinco.getCincominutalId(), est);  
  
 //Single  
 dom.setTipoCalculo("single");  
 dom.setCincominutalEST(est);  
 DominioModeloMatematico response = calculaFactores(dom);  
 DominioEST estResponse = response.getCincominutalEST();  
 List<TxCincominutalVar> listaSaves = new ArrayList<>();  
 System.out.println("CpwAT calculado: " + estResponse.getCpwAT());  
 System.out.println("variables a actualizar: " + cincominutalVarByCincominutal.size());  
 for (TxCincominutalVar txVar : cincominutalVarByCincominutal) {  
 int varId = txVar.getCatVariable()  
 .getVariableId()  
 .intValue();  
 if (varId == 1) {  
 txVar.setFcPrimera(estResponse.getCpwAT());  
 }  
 if (varId == 3) {  
 txVar.setFcPrimera(estResponse.getCpwBP());  
 }  
 if (varId == 2) {  
 txVar.setFcPrimera(estResponse.getCpwRH());  
 }  
 if (varId == 4) {  
 txVar.setFcPrimera(estResponse.getCpwPCI());  
 }  
 if (varId == 12) {  
 txVar.setFcPrimera(estResponse.getCpwCW());  
 }  
 if (varId == 17) {  
 txVar.setFcPrimera(estResponse.getCpwFP());  
 }  
 txVar.setFechaPc(new Date());  
 if (obj.getTipoEnergia().equals("ES")) { //Estación de supervisión  
 //cinco.setEnergiaPc(estResponse.getSummerCapacity());//La tabla TX\_CINCOMINUTAL debe tener la propiedad energiaPc  
 //daoTxCincominutal.save(cinco);  
  
  
  
  
  
  
  
  
  
  
  
  
 }  
 if (obj.getTipoEnergia()  
 .equals("PERFILES")) { //Perfiles de energis  
 TxPerfilesVariable perfil =  
 daoTxPerfilesVariable.getPerfilForMM(cinco.getCincominutalId());  
 perfil.setEnergiaPc(estResponse.getSummerCapacity());  
 perfil.setFechaPc(new Date());  
 daoTxPerfilesVariable.save(perfil);  
 }  
 listaSaves.add(txVar);  
 }  
 daoCincoVar.saveAll(listaSaves);  
 }  
 cincos.add(cinco);  
 }  
 // dom.setListaEST(aux);  
 //Batch  
 //dom.setTipoCalculo("batch");  
 /\*DominioModeloMatematico response = calculaFactores(dom);  
 for(BigDecimal idCinco: mapa.keySet()) {  
  
 }\*/  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 List<TxCincominutal> cincos;  
  
 private boolean existCinco(TxCincominutal txCinco) {  
 for (TxCincominutal obj : cincos) {  
 if (obj.getCincominutalId().intValue() == txCinco.getCincominutalId().intValue()) {  
 return true;  
 }  
 }  
 return false;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\mm\IfaceModeloMatematico.java

package mx.mbi.dgc.business.mm;  
  
import java.util.List;  
  
import mx.mbi.dgc.dominio.DominioDyCNorma;  
import mx.mbi.dgc.dominio.negocio.DominioModeloMatematico;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxCincominutalVar;  
  
public interface IfaceModeloMatematico {  
 DominioModeloMatematico calculaFactores(DominioModeloMatematico dom) throws DyCNormaException;  
 void actualizaMM(DominioDyCNorma obj, List<TxCincominutalVar> lista) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\negocio\BusinessCapturaValoresCargos.java

package mx.mbi.dgc.business.negocio;  
  
import java.math.BigDecimal;  
  
import java.util.ArrayList;  
import java.util.Date;  
  
import java.util.List;  
  
import mx.mbi.dgc.dao.DAOCatPlanta;  
import mx.mbi.dgc.dao.negocio.DAOCargoFijo;  
import mx.mbi.dgc.dao.negocio.DAOCargos;  
import mx.mbi.dgc.dominio.DominioCatCargoFijo;  
import mx.mbi.dgc.dominio.DominioCatCargos;  
import mx.mbi.dgc.dominio.DominioCatSubCargos;  
import mx.mbi.dgc.dominio.negocio.DominioCapturaValoresCargos;  
import mx.mbi.dgc.dominio.negocio.DominioDetalleCVC;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
  
import mx.mbi.dgc.jpa.entity.CatCargoFijo;  
import mx.mbi.dgc.jpa.entity.CatCargos;  
  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.jpa.entity.CatSubCargos;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BusinessCapturaValoresCargos implements IfaceCapturaValoresCargos {  
 private static final Logger LOG = LogManager.getLogger(BusinessCapturaValoresCargos.class);  
 @Autowired  
 private DAOCargos daoCargos;  
 @Autowired  
 private DAOCargoFijo daoCatCargoFijo;  
 @Autowired  
 private DAOCatPlanta daoCatPlanta;  
   
 @Override  
 public DominioCapturaValoresCargos getCargos(BigDecimal idPlanta) throws DyCNormaException {  
 LOG.info("#### BusinessCapturaValoresCargos.getCargos: "+new Date());  
 DominioCapturaValoresCargos response = new DominioCapturaValoresCargos();  
 response.setStatus(false);  
 response.setCode(500);  
 try {  
 CatPlanta catPlanta = daoCatPlanta.findById(idPlanta).orElse(null);  
 List<DominioCatCargos> listaCargos = new ArrayList<>();  
 List<CatCargos> lista = daoCargos.findByCatPlanta(catPlanta);  
 for(CatCargos catCargo: lista) {  
 DominioCatCargos domCargo = new DominioCatCargos();  
 JRUtil.copyProperties(domCargo, catCargo);  
 List<DominioCatSubCargos> listaDominioCatSubCargos = new ArrayList<>();  
 for(CatSubCargos subCargo: catCargo.getCatSubCargoses()) {  
 DominioCatSubCargos domSubCargo = new DominioCatSubCargos();  
 JRUtil.copyProperties(domSubCargo, subCargo);  
 List<DominioCatCargoFijo> listaDominioCatCargoFijo = new ArrayList<>();  
 for(CatCargoFijo catCargoFijo: subCargo.getCatCargoFijos()) {  
 if(catCargoFijo.isPrincipal()) {  
 DominioCatCargoFijo domCatCargoFijo = new DominioCatCargoFijo();  
 JRUtil.copyProperties(domCatCargoFijo, catCargoFijo);  
 listaDominioCatCargoFijo.add(domCatCargoFijo);  
 }  
 }  
 domSubCargo.setCatCargoFijoList(listaDominioCatCargoFijo);  
 listaDominioCatSubCargos.add(domSubCargo);  
 }  
 domCargo.setCatSubCargosList(listaDominioCatSubCargos);  
 listaCargos.add(domCargo);  
 }  
 response.setListaCargos(listaCargos);  
 response.setStatus(true);  
 response.setCode(200);  
 } catch(Exception ex) {  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en DominioCapturaValoresCargos.getCargos: "+ex);  
 }  
 return response;  
 }  
  
 @Override  
 public DominioCapturaValoresCargos cargaEvidencia(DominioCapturaValoresCargos dom) throws DyCNormaException {  
 LOG.info("#### BusinessCapturaValoresCargos.cargaEvidencia: "+new Date());  
 //System.out.println("######### dom.getCargoFijoId(): "+dom.getCargoFijoId());  
 //System.out.println("######### dom.getPathSoporte(): "+dom.getPathSoporte());  
 System.out.println("######### dom.getValor(): "+dom.getValor());  
 try {  
 CatCargoFijo catCargoFijo = daoCatCargoFijo.findById(dom.getCargoFijoId()).orElse(null);  
 CatCargoFijo nuevoCargoFijo = new CatCargoFijo();  
 JRUtil.copyProperties(nuevoCargoFijo, catCargoFijo);  
 //System.out.println("######### dom.getValor(): "+dom.getValor());  
 nuevoCargoFijo.setCargoFijoId(new BigDecimal(-1));  
 nuevoCargoFijo.setPathSoporte(dom.getPathSoporte());  
 nuevoCargoFijo.setNombreEvidencia(dom.getNombreEvidencia());  
 nuevoCargoFijo.setValor(dom.getValor());  
 nuevoCargoFijo.setPrincipal(true);  
   
 catCargoFijo.setPrincipal(false);  
 daoCatCargoFijo.save(catCargoFijo);  
 daoCatCargoFijo.save(nuevoCargoFijo);  
   
 dom.setStatus(true);  
 dom.setCode(200);  
 //System.out.println("######### ID nuevoCargoFijo: "+nuevoCargoFijo.getCargoFijoId());  
 } catch(Exception ex) {  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en DominioCapturaValoresCargos.cargaEvidencia: "+ex);  
 }  
 return dom;  
 }  
  
 @Override  
 public DominioCapturaValoresCargos getDetalle(DominioCapturaValoresCargos dom) throws DyCNormaException {  
 LOG.info("#### BusinessCapturaValoresCargos.getDetalle: "+new Date());  
 //System.out.println("######### dom.getCargoFijoId(): "+dom.getCargoFijoId());  
 try {  
 CatCargoFijo catCargoFijo = daoCatCargoFijo.findById(dom.getCargoFijoId()).orElse(null);  
 List<DominioDetalleCVC> detalleCVC = new ArrayList<>();  
 List<CatCargoFijo> listaDetalle = daoCatCargoFijo.findByCatMesCargoAndCatSubCargos(  
 catCargoFijo.getCatMesCargo(), catCargoFijo.getCatSubCargos());  
 for(CatCargoFijo obj: listaDetalle) {  
 DominioDetalleCVC domDetalleCVC = new DominioDetalleCVC();  
 domDetalleCVC.setCargoFijoId(obj.getCargoFijoId());  
 domDetalleCVC.setNombreEvidencia(obj.getNombreEvidencia());  
 domDetalleCVC.setPathSoporte(obj.getPathSoporte());  
 domDetalleCVC.setValor(obj.getValor());  
 detalleCVC.add(domDetalleCVC);  
 }  
 dom.setDetalleCVC(detalleCVC);  
 dom.setStatus(true);  
 dom.setCode(200);  
 //System.out.println("######### dom.getDetalleCVC.size(): "+dom.getDetalleCVC().size());  
 } catch(Exception ex) {  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en DominioCapturaValoresCargos.getDetalle: "+ex);  
 }  
 return dom;  
 }  
  
 @Override  
 public DominioDetalleCVC getFilePath(DominioDetalleCVC dom) throws DyCNormaException {  
 LOG.info("#### BusinessCapturaValoresCargos.getFilePath: ");  
 //System.out.println("######### dom.getCargoFijoId(): "+dom.getCargoFijoId());  
 try {  
 CatCargoFijo catCargoFijo = daoCatCargoFijo.findById(dom.getCargoFijoId()).orElse(null);  
 dom.setNombreEvidencia(catCargoFijo.getNombreEvidencia());  
 dom.setPathSoporte(catCargoFijo.getPathSoporte());  
 } catch(Exception ex) {  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en DominioCapturaValoresCargos.getFilePath: "+ex);  
 }  
 return dom;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\business\negocio\IfaceCapturaValoresCargos.java

package mx.mbi.dgc.business.negocio;  
  
import java.math.BigDecimal;  
  
import mx.mbi.dgc.dominio.negocio.DominioCapturaValoresCargos;  
import mx.mbi.dgc.dominio.negocio.DominioDetalleCVC;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
  
public interface IfaceCapturaValoresCargos {  
 DominioCapturaValoresCargos getCargos(BigDecimal idPlanta) throws DyCNormaException;  
 DominioCapturaValoresCargos cargaEvidencia(DominioCapturaValoresCargos dom) throws DyCNormaException;  
 DominioCapturaValoresCargos getDetalle(DominioCapturaValoresCargos dom) throws DyCNormaException;  
 DominioDetalleCVC getFilePath(DominioDetalleCVC dom) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\eat\ArrayOfEATInput.java

package mx.mbi.dgc.client.exe.eat;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para ArrayOfEATInput complex type.  
 \*  
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*  
 \* <pre>  
 \* &lt;complexType name="ArrayOfEATInput">  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="EATInput" type="{http://172.20.141.102/DGCeat/api/eatmmodel.asmx}EATInput" maxOccurs="unbounded" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*  
 \*  
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "ArrayOfEATInput", propOrder = {  
 "eatInput"  
})  
public class ArrayOfEATInput {  
  
 @XmlElement(name = "EATInput", nillable = true)  
 protected List<EATInput> eatInput;  
  
 /\*\*  
 \* Gets the value of the eatInput property.  
 \*   
 \* <p>  
 \* This accessor method returns a reference to the live list,  
 \* not a snapshot. Therefore any modification you make to the  
 \* returned list will be present inside the JAXB object.  
 \* This is why there is not a <CODE>set</CODE> method for the eatInput property.  
 \*   
 \* <p>  
 \* For example, to add a new item, do as follows:  
 \* <pre>  
 \* getEATInput().add(newItem);  
 \* </pre>  
 \*   
 \*   
 \* <p>  
 \* Objects of the following type(s) are allowed in the list  
 \* {@link EATInput }  
 \*   
 \*   
 \*/  
 public List<EATInput> getEATInput() {  
 if (eatInput == null) {  
 eatInput = new ArrayList<EATInput>();  
 }  
 return this.eatInput;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\eat\ArrayOfEATOutput.java

package mx.mbi.dgc.client.exe.eat;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para ArrayOfEATOutput complex type.  
 \*  
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*  
 \* <pre>  
 \* &lt;complexType name="ArrayOfEATOutput">  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="EATOutput" type="{http://172.20.141.102/DGCeat/api/eatmmodel.asmx}EATOutput" maxOccurs="unbounded" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*  
 \*  
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "ArrayOfEATOutput", propOrder = {  
 "eatOutput"  
})  
public class ArrayOfEATOutput {  
  
 @XmlElement(name = "EATOutput", nillable = true)  
 protected List<EATOutput> eatOutput;  
  
 /\*\*  
 \* Gets the value of the eatOutput property.  
 \*   
 \* <p>  
 \* This accessor method returns a reference to the live list,  
 \* not a snapshot. Therefore any modification you make to the  
 \* returned list will be present inside the JAXB object.  
 \* This is why there is not a <CODE>set</CODE> method for the eatOutput property.  
 \*   
 \* <p>  
 \* For example, to add a new item, do as follows:  
 \* <pre>  
 \* getEATOutput().add(newItem);  
 \* </pre>  
 \*   
 \*   
 \* <p>  
 \* Objects of the following type(s) are allowed in the list  
 \* {@link EATOutput }  
 \*   
 \*   
 \*/  
 public List<EATOutput> getEATOutput() {  
 if (eatOutput == null) {  
 eatOutput = new ArrayList<EATOutput>();  
 }  
 return this.eatOutput;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\eat\EATBatch.java

package mx.mbi.dgc.client.exe.eat;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para anonymous complex type.  
 \*  
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*  
 \* <pre>  
 \* &lt;complexType>  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="inputs" type="{http://172.20.141.102/DGCeat/api/eatmmodel.asmx}ArrayOfEATInput" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*  
 \*  
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "", propOrder = {  
 "inputs"  
})  
@XmlRootElement(name = "EATBatch")  
public class EATBatch {  
  
 protected ArrayOfEATInput inputs;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad inputs.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link ArrayOfEATInput }  
 \*   
 \*/  
 public ArrayOfEATInput getInputs() {  
 return inputs;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad inputs.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link ArrayOfEATInput }  
 \*   
 \*/  
 public void setInputs(ArrayOfEATInput value) {  
 this.inputs = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\eat\EATBatchResponse.java

package mx.mbi.dgc.client.exe.eat;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para anonymous complex type.  
 \*  
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*  
 \* <pre>  
 \* &lt;complexType>  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="EATBatchResult" type="{http://172.20.141.102/DGCeat/api/eatmmodel.asmx}ArrayOfEATOutput" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*  
 \*  
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "", propOrder = {  
 "eatBatchResult"  
})  
@XmlRootElement(name = "EATBatchResponse")  
public class EATBatchResponse {  
  
 @XmlElement(name = "EATBatchResult")  
 protected ArrayOfEATOutput eatBatchResult;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad eatBatchResult.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link ArrayOfEATOutput }  
 \*   
 \*/  
 public ArrayOfEATOutput getEATBatchResult() {  
 return eatBatchResult;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad eatBatchResult.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link ArrayOfEATOutput }  
 \*   
 \*/  
 public void setEATBatchResult(ArrayOfEATOutput value) {  
 this.eatBatchResult = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\eat\EATInput.java

package mx.mbi.dgc.client.exe.eat;  
  
import java.math.BigDecimal;  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlSeeAlso;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para EATInput complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType name="EATInput">  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="FuelType" type="{http://www.w3.org/2001/XMLSchema}int"/>  
 \* &lt;element name="CurrentCapacity" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="ManualNetCapacity" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="DryBulbTemp" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="AtmosphericPressure" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="HumidityRelative" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="CalorificPower" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="CapacitorPressure" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="PowerFactor" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="ProvenNetCapacityGas" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="ProvenNetCapacityOil" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="AvailableEnergy" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "EATInput", propOrder = {  
 "fuelType",  
 "currentCapacity",  
 "manualNetCapacity",  
 "dryBulbTemp",  
 "atmosphericPressure",  
 "humidityRelative",  
 "calorificPower",  
 "capacitorPressure",  
 "powerFactor",  
 "provenNetCapacityGas",  
 "provenNetCapacityOil",  
 "availableEnergy"  
})  
@XmlSeeAlso({  
 EATOutput.class  
})  
public class EATInput {  
  
 @XmlElement(name = "FuelType")  
 protected int fuelType;  
 @XmlElement(name = "CurrentCapacity", required = true)  
 protected BigDecimal currentCapacity;  
 @XmlElement(name = "ManualNetCapacity", required = true)  
 protected BigDecimal manualNetCapacity;  
 @XmlElement(name = "DryBulbTemp", required = true)  
 protected BigDecimal dryBulbTemp;  
 @XmlElement(name = "AtmosphericPressure", required = true)  
 protected BigDecimal atmosphericPressure;  
 @XmlElement(name = "HumidityRelative", required = true)  
 protected BigDecimal humidityRelative;  
 @XmlElement(name = "CalorificPower", required = true)  
 protected BigDecimal calorificPower;  
 @XmlElement(name = "CapacitorPressure", required = true)  
 protected BigDecimal capacitorPressure;  
 @XmlElement(name = "PowerFactor", required = true)  
 protected BigDecimal powerFactor;  
 @XmlElement(name = "ProvenNetCapacityGas", required = true)  
 protected BigDecimal provenNetCapacityGas;  
 @XmlElement(name = "ProvenNetCapacityOil", required = true)  
 protected BigDecimal provenNetCapacityOil;  
 @XmlElement(name = "AvailableEnergy", required = true)  
 protected BigDecimal availableEnergy;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad fuelType.  
 \*   
 \*/  
 public int getFuelType() {  
 return fuelType;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad fuelType.  
 \*   
 \*/  
 public void setFuelType(int value) {  
 this.fuelType = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad currentCapacity.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCurrentCapacity() {  
 return currentCapacity;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad currentCapacity.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCurrentCapacity(BigDecimal value) {  
 this.currentCapacity = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad manualNetCapacity.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getManualNetCapacity() {  
 return manualNetCapacity;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad manualNetCapacity.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setManualNetCapacity(BigDecimal value) {  
 this.manualNetCapacity = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad dryBulbTemp.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getDryBulbTemp() {  
 return dryBulbTemp;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad dryBulbTemp.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setDryBulbTemp(BigDecimal value) {  
 this.dryBulbTemp = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad atmosphericPressure.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getAtmosphericPressure() {  
 return atmosphericPressure;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad atmosphericPressure.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setAtmosphericPressure(BigDecimal value) {  
 this.atmosphericPressure = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad humidityRelative.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getHumidityRelative() {  
 return humidityRelative;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad humidityRelative.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setHumidityRelative(BigDecimal value) {  
 this.humidityRelative = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad calorificPower.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCalorificPower() {  
 return calorificPower;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad calorificPower.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCalorificPower(BigDecimal value) {  
 this.calorificPower = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad capacitorPressure.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCapacitorPressure() {  
 return capacitorPressure;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad capacitorPressure.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCapacitorPressure(BigDecimal value) {  
 this.capacitorPressure = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad powerFactor.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getPowerFactor() {  
 return powerFactor;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad powerFactor.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setPowerFactor(BigDecimal value) {  
 this.powerFactor = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad provenNetCapacityGas.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getProvenNetCapacityGas() {  
 return provenNetCapacityGas;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad provenNetCapacityGas.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setProvenNetCapacityGas(BigDecimal value) {  
 this.provenNetCapacityGas = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad provenNetCapacityOil.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getProvenNetCapacityOil() {  
 return provenNetCapacityOil;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad provenNetCapacityOil.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setProvenNetCapacityOil(BigDecimal value) {  
 this.provenNetCapacityOil = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad availableEnergy.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getAvailableEnergy() {  
 return availableEnergy;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad availableEnergy.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setAvailableEnergy(BigDecimal value) {  
 this.availableEnergy = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\eat\EATMModel.java

package mx.mbi.dgc.client.exe.eat;  
  
import java.net.MalformedURLException;  
import java.net.URL;  
import javax.xml.namespace.QName;  
import javax.xml.ws.Service;  
import javax.xml.ws.WebEndpoint;  
import javax.xml.ws.WebServiceClient;  
import javax.xml.ws.WebServiceException;  
import javax.xml.ws.WebServiceFeature;  
  
  
/\*\*  
 \* This class was generated by the JAX-WS RI.  
 \* JAX-WS RI 2.2.8  
 \* Generated source version: 2.2  
 \*   
 \*/  
@WebServiceClient(name = "EATMModel", targetNamespace = "http://172.20.141.102/DGCeat/api/eatmmodel.asmx", wsdlLocation = "http://172.20.141.102/DGCeat/api/eatmmodel.asmx?wsdl")  
public class EATMModel  
 extends Service  
{  
  
 private final static URL EATMMODEL\_WSDL\_LOCATION;  
 private final static WebServiceException EATMMODEL\_EXCEPTION;  
 private final static QName EATMMODEL\_QNAME = new QName("http://172.20.141.102/DGCeat/api/eatmmodel.asmx", "EATMModel");  
  
 static {  
 URL url = null;  
 WebServiceException e = null;  
 try {  
 url = new URL("http://172.20.141.102/DGCeat/api/eatmmodel.asmx?wsdl");  
 } catch (MalformedURLException ex) {  
 e = new WebServiceException(ex);  
 }  
 EATMMODEL\_WSDL\_LOCATION = url;  
 EATMMODEL\_EXCEPTION = e;  
 }  
  
 public EATMModel() {  
 super(\_\_getWsdlLocation(), EATMMODEL\_QNAME);  
 }  
  
 public EATMModel(WebServiceFeature... features) {  
 super(\_\_getWsdlLocation(), EATMMODEL\_QNAME, features);  
 }  
  
 public EATMModel(URL wsdlLocation) {  
 super(wsdlLocation, EATMMODEL\_QNAME);  
 }  
  
 public EATMModel(URL wsdlLocation, WebServiceFeature... features) {  
 super(wsdlLocation, EATMMODEL\_QNAME, features);  
 }  
  
 public EATMModel(URL wsdlLocation, QName serviceName) {  
 super(wsdlLocation, serviceName);  
 }  
  
 public EATMModel(URL wsdlLocation, QName serviceName, WebServiceFeature... features) {  
 super(wsdlLocation, serviceName, features);  
 }  
  
 /\*\*  
 \*   
 \* @return  
 \* returns EATMModelSoap  
 \*/  
 @WebEndpoint(name = "EATMModelSoap")  
 public EATMModelSoap getEATMModelSoap() {  
 return super.getPort(new QName("http://172.20.141.102/DGCeat/api/eatmmodel.asmx", "EATMModelSoap"), EATMModelSoap.class);  
 }  
  
 /\*\*  
 \*   
 \* @param features  
 \* A list of {@link javax.xml.ws.WebServiceFeature} to configure on the proxy. Supported features not in the <code>features</code> parameter will have their default values.  
 \* @return  
 \* returns EATMModelSoap  
 \*/  
 @WebEndpoint(name = "EATMModelSoap")  
 public EATMModelSoap getEATMModelSoap(WebServiceFeature... features) {  
 return super.getPort(new QName("http://172.20.141.102/DGCeat/api/eatmmodel.asmx", "EATMModelSoap"), EATMModelSoap.class, features);  
 }  
  
 /\*\*  
 \*   
 \* @return  
 \* returns EATMModelSoap  
 \*/  
 @WebEndpoint(name = "EATMModelSoap12")  
 public EATMModelSoap getEATMModelSoap12() {  
 return super.getPort(new QName("http://172.20.141.102/DGCeat/api/eatmmodel.asmx", "EATMModelSoap12"), EATMModelSoap.class);  
 }  
  
 /\*\*  
 \*   
 \* @param features  
 \* A list of {@link javax.xml.ws.WebServiceFeature} to configure on the proxy. Supported features not in the <code>features</code> parameter will have their default values.  
 \* @return  
 \* returns EATMModelSoap  
 \*/  
 @WebEndpoint(name = "EATMModelSoap12")  
 public EATMModelSoap getEATMModelSoap12(WebServiceFeature... features) {  
 return super.getPort(new QName("http://172.20.141.102/DGCeat/api/eatmmodel.asmx", "EATMModelSoap12"), EATMModelSoap.class, features);  
 }  
  
 private static URL \_\_getWsdlLocation() {  
 if (EATMMODEL\_EXCEPTION!= null) {  
 throw EATMMODEL\_EXCEPTION;  
 }  
 return EATMMODEL\_WSDL\_LOCATION;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\eat\EATMModelSoap.java

package mx.mbi.dgc.client.exe.eat;  
  
import javax.jws.WebMethod;  
import javax.jws.WebParam;  
import javax.jws.WebResult;  
import javax.jws.WebService;  
import javax.xml.bind.annotation.XmlSeeAlso;  
import javax.xml.ws.RequestWrapper;  
import javax.xml.ws.ResponseWrapper;  
  
  
/\*\*  
 \* This class was generated by the JAX-WS RI.  
 \* JAX-WS RI 2.2.8  
 \* Generated source version: 2.2  
 \*   
 \*/  
@WebService(name = "EATMModelSoap", targetNamespace = "http://172.20.141.102/DGCeat/api/eatmmodel.asmx")  
@XmlSeeAlso({  
 ObjectFactory.class  
})  
public interface EATMModelSoap {  
  
  
 /\*\*  
 \*   
 \* @param input  
 \* @return  
 \* returns mx.mbi.dgc.client.exe.eat.EATOutput  
 \*/  
 @WebMethod(operationName = "EATSingle", action = "http://172.20.141.102/DGCeat/api/eatmmodel.asmx/EATSingle")  
 @WebResult(name = "EATSingleResult", targetNamespace = "http://172.20.141.102/DGCeat/api/eatmmodel.asmx")  
 @RequestWrapper(localName = "EATSingle", targetNamespace = "http://172.20.141.102/DGCeat/api/eatmmodel.asmx", className = "mx.mbi.dgc.client.exe.eat.EATSingle")  
 @ResponseWrapper(localName = "EATSingleResponse", targetNamespace = "http://172.20.141.102/DGCeat/api/eatmmodel.asmx", className = "mx.mbi.dgc.client.exe.eat.EATSingleResponse")  
 public EATOutput eatSingle(  
 @WebParam(name = "input", targetNamespace = "http://172.20.141.102/DGCeat/api/eatmmodel.asmx")  
 EATInput input);  
  
 /\*\*  
 \*   
 \* @param inputs  
 \* @return  
 \* returns mx.mbi.dgc.client.exe.eat.ArrayOfEATOutput  
 \*/  
 @WebMethod(operationName = "EATBatch", action = "http://172.20.141.102/DGCeat/api/eatmmodel.asmx/EATBatch")  
 @WebResult(name = "EATBatchResult", targetNamespace = "http://172.20.141.102/DGCeat/api/eatmmodel.asmx")  
 @RequestWrapper(localName = "EATBatch", targetNamespace = "http://172.20.141.102/DGCeat/api/eatmmodel.asmx", className = "mx.mbi.dgc.client.exe.eat.EATBatch")  
 @ResponseWrapper(localName = "EATBatchResponse", targetNamespace = "http://172.20.141.102/DGCeat/api/eatmmodel.asmx", className = "mx.mbi.dgc.client.exe.eat.EATBatchResponse")  
 public ArrayOfEATOutput eatBatch(  
 @WebParam(name = "inputs", targetNamespace = "http://172.20.141.102/DGCeat/api/eatmmodel.asmx")  
 ArrayOfEATInput inputs);  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\eat\EATOutput.java

package mx.mbi.dgc.client.exe.eat;  
  
import java.math.BigDecimal;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para EATOutput complex type.  
 \*  
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*  
 \* <pre>  
 \* &lt;complexType name="EATOutput">  
 \* &lt;complexContent>  
 \* &lt;extension base="{http://172.20.141.102/DGCeat/api/eatmmodel.asmx}EATInput">  
 \* &lt;sequence>  
 \* &lt;element name="CurrenPlantLoad" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="NetCapacity" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="SummerCapacity" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="CTUNGLoad" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>  
 \* &lt;element name="CTUNG" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>  
 \* &lt;element name="Cpw\_AT" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Cpw\_BP" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Cpw\_RH" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Cpw\_LHV" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Cpw\_VC" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Cpw\_PF" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Chr\_AT" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Chr\_BP" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Chr\_RH" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Chr\_LHV" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Chr\_VC" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Chr\_PF" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;/sequence>  
 \* &lt;/extension>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*  
 \*  
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "EATOutput", propOrder = {  
 "currenPlantLoad",  
 "netCapacity",  
 "summerCapacity",  
 "ctungLoad",  
 "ctung",  
 "cpwAT",  
 "cpwBP",  
 "cpwRH",  
 "cpwLHV",  
 "cpwVC",  
 "cpwPF",  
 "chrAT",  
 "chrBP",  
 "chrRH",  
 "chrLHV",  
 "chrVC",  
 "chrPF"  
})  
public class EATOutput  
 extends EATInput  
{  
  
 @XmlElement(name = "CurrenPlantLoad", required = true)  
 protected BigDecimal currenPlantLoad;  
 @XmlElement(name = "NetCapacity", required = true)  
 protected BigDecimal netCapacity;  
 @XmlElement(name = "SummerCapacity", required = true)  
 protected BigDecimal summerCapacity;  
 @XmlElement(name = "CTUNGLoad")  
 protected String ctungLoad;  
 @XmlElement(name = "CTUNG")  
 protected String ctung;  
 @XmlElement(name = "Cpw\_AT", required = true)  
 protected BigDecimal cpwAT;  
 @XmlElement(name = "Cpw\_BP", required = true)  
 protected BigDecimal cpwBP;  
 @XmlElement(name = "Cpw\_RH", required = true)  
 protected BigDecimal cpwRH;  
 @XmlElement(name = "Cpw\_LHV", required = true)  
 protected BigDecimal cpwLHV;  
 @XmlElement(name = "Cpw\_VC", required = true)  
 protected BigDecimal cpwVC;  
 @XmlElement(name = "Cpw\_PF", required = true)  
 protected BigDecimal cpwPF;  
 @XmlElement(name = "Chr\_AT", required = true)  
 protected BigDecimal chrAT;  
 @XmlElement(name = "Chr\_BP", required = true)  
 protected BigDecimal chrBP;  
 @XmlElement(name = "Chr\_RH", required = true)  
 protected BigDecimal chrRH;  
 @XmlElement(name = "Chr\_LHV", required = true)  
 protected BigDecimal chrLHV;  
 @XmlElement(name = "Chr\_VC", required = true)  
 protected BigDecimal chrVC;  
 @XmlElement(name = "Chr\_PF", required = true)  
 protected BigDecimal chrPF;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad currenPlantLoad.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCurrenPlantLoad() {  
 return currenPlantLoad;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad currenPlantLoad.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCurrenPlantLoad(BigDecimal value) {  
 this.currenPlantLoad = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad netCapacity.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getNetCapacity() {  
 return netCapacity;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad netCapacity.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setNetCapacity(BigDecimal value) {  
 this.netCapacity = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad summerCapacity.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getSummerCapacity() {  
 return summerCapacity;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad summerCapacity.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setSummerCapacity(BigDecimal value) {  
 this.summerCapacity = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad ctungLoad.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getCTUNGLoad() {  
 return ctungLoad;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad ctungLoad.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setCTUNGLoad(String value) {  
 this.ctungLoad = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad ctung.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getCTUNG() {  
 return ctung;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad ctung.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setCTUNG(String value) {  
 this.ctung = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad cpwAT.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCpwAT() {  
 return cpwAT;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad cpwAT.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCpwAT(BigDecimal value) {  
 this.cpwAT = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad cpwBP.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCpwBP() {  
 return cpwBP;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad cpwBP.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCpwBP(BigDecimal value) {  
 this.cpwBP = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad cpwRH.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCpwRH() {  
 return cpwRH;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad cpwRH.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCpwRH(BigDecimal value) {  
 this.cpwRH = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad cpwLHV.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCpwLHV() {  
 return cpwLHV;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad cpwLHV.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCpwLHV(BigDecimal value) {  
 this.cpwLHV = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad cpwVC.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCpwVC() {  
 return cpwVC;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad cpwVC.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCpwVC(BigDecimal value) {  
 this.cpwVC = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad cpwPF.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCpwPF() {  
 return cpwPF;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad cpwPF.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCpwPF(BigDecimal value) {  
 this.cpwPF = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad chrAT.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getChrAT() {  
 return chrAT;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad chrAT.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setChrAT(BigDecimal value) {  
 this.chrAT = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad chrBP.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getChrBP() {  
 return chrBP;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad chrBP.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setChrBP(BigDecimal value) {  
 this.chrBP = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad chrRH.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getChrRH() {  
 return chrRH;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad chrRH.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setChrRH(BigDecimal value) {  
 this.chrRH = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad chrLHV.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getChrLHV() {  
 return chrLHV;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad chrLHV.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setChrLHV(BigDecimal value) {  
 this.chrLHV = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad chrVC.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getChrVC() {  
 return chrVC;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad chrVC.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setChrVC(BigDecimal value) {  
 this.chrVC = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad chrPF.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getChrPF() {  
 return chrPF;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad chrPF.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setChrPF(BigDecimal value) {  
 this.chrPF = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\eat\EATSingle.java

package mx.mbi.dgc.client.exe.eat;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para anonymous complex type.  
 \*  
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*  
 \* <pre>  
 \* &lt;complexType>  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="input" type="{http://172.20.141.102/DGCeat/api/eatmmodel.asmx}EATInput" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*  
 \*  
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "", propOrder = {  
 "input"  
})  
@XmlRootElement(name = "EATSingle")  
public class EATSingle {  
  
 protected EATInput input;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad input.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link EATInput }  
 \*   
 \*/  
 public EATInput getInput() {  
 return input;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad input.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link EATInput }  
 \*   
 \*/  
 public void setInput(EATInput value) {  
 this.input = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\eat\EATSingleResponse.java

package mx.mbi.dgc.client.exe.eat;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para anonymous complex type.  
 \*  
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*  
 \* <pre>  
 \* &lt;complexType>  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="EATSingleResult" type="{http://172.20.141.102/DGCeat/api/eatmmodel.asmx}EATOutput" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*  
 \*  
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "", propOrder = {  
 "eatSingleResult"  
})  
@XmlRootElement(name = "EATSingleResponse")  
public class EATSingleResponse {  
  
 @XmlElement(name = "EATSingleResult")  
 protected EATOutput eatSingleResult;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad eatSingleResult.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link EATOutput }  
 \*   
 \*/  
 public EATOutput getEATSingleResult() {  
 return eatSingleResult;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad eatSingleResult.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link EATOutput }  
 \*   
 \*/  
 public void setEATSingleResult(EATOutput value) {  
 this.eatSingleResult = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\eat\ObjectFactory.java

package mx.mbi.dgc.client.exe.eat;  
  
import javax.xml.bind.annotation.XmlRegistry;  
  
  
/\*\*  
 \* This object contains factory methods for each   
 \* Java content interface and Java element interface   
 \* generated in the mx.mbi.dgc.client.exe.eat package.   
 \* <p>An ObjectFactory allows you to programatically   
 \* construct new instances of the Java representation   
 \* for XML content. The Java representation of XML   
 \* content can consist of schema derived interfaces   
 \* and classes representing the binding of schema   
 \* type definitions, element declarations and model   
 \* groups. Factory methods for each of these are   
 \* provided in this class.  
 \*   
 \*/  
@XmlRegistry  
public class ObjectFactory {  
  
  
 /\*\*  
 \* Create a new ObjectFactory that can be used to create new instances of schema derived classes for package: mx.mbi.dgc.client.exe.eat  
 \*   
 \*/  
 public ObjectFactory() {  
 }  
  
 /\*\*  
 \* Create an instance of {@link EATSingle }  
 \*   
 \*/  
 public EATSingle createEATSingle() {  
 return new EATSingle();  
 }  
  
 /\*\*  
 \* Create an instance of {@link EATInput }  
 \*   
 \*/  
 public EATInput createEATInput() {  
 return new EATInput();  
 }  
  
 /\*\*  
 \* Create an instance of {@link EATBatchResponse }  
 \*   
 \*/  
 public EATBatchResponse createEATBatchResponse() {  
 return new EATBatchResponse();  
 }  
  
 /\*\*  
 \* Create an instance of {@link ArrayOfEATOutput }  
 \*   
 \*/  
 public ArrayOfEATOutput createArrayOfEATOutput() {  
 return new ArrayOfEATOutput();  
 }  
  
 /\*\*  
 \* Create an instance of {@link EATSingleResponse }  
 \*   
 \*/  
 public EATSingleResponse createEATSingleResponse() {  
 return new EATSingleResponse();  
 }  
  
 /\*\*  
 \* Create an instance of {@link EATOutput }  
 \*   
 \*/  
 public EATOutput createEATOutput() {  
 return new EATOutput();  
 }  
  
 /\*\*  
 \* Create an instance of {@link EATBatch }  
 \*   
 \*/  
 public EATBatch createEATBatch() {  
 return new EATBatch();  
 }  
  
 /\*\*  
 \* Create an instance of {@link ArrayOfEATInput }  
 \*   
 \*/  
 public ArrayOfEATInput createArrayOfEATInput() {  
 return new ArrayOfEATInput();  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\eat\package-info.java

@javax.xml.bind.annotation.XmlSchema(namespace = "http://172.20.141.102/DGCeat/api/eatmmodel.asmx", elementFormDefault = javax.xml.bind.annotation.XmlNsForm.QUALIFIED)  
package mx.mbi.dgc.client.exe.eat;

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\est\ArrayOfESTInput.java

package mx.mbi.dgc.client.exe.est;  
  
import java.util.ArrayList;  
import java.util.List;  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para ArrayOfESTInput complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType name="ArrayOfESTInput">  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="ESTInput" type="{http://172.20.141.102/DGCest/api/estmmodel.asmx}ESTInput" maxOccurs="unbounded" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "ArrayOfESTInput", propOrder = {  
 "estInput"  
})  
public class ArrayOfESTInput {  
  
 @XmlElement(name = "ESTInput", nillable = true)  
 protected List<ESTInput> estInput;  
  
 /\*\*  
 \* Gets the value of the estInput property.  
 \*   
 \* <p>  
 \* This accessor method returns a reference to the live list,  
 \* not a snapshot. Therefore any modification you make to the  
 \* returned list will be present inside the JAXB object.  
 \* This is why there is not a <CODE>set</CODE> method for the estInput property.  
 \*   
 \* <p>  
 \* For example, to add a new item, do as follows:  
 \* <pre>  
 \* getESTInput().add(newItem);  
 \* </pre>  
 \*   
 \*   
 \* <p>  
 \* Objects of the following type(s) are allowed in the list  
 \* {@link ESTInput }  
 \*   
 \*   
 \*/  
 public List<ESTInput> getESTInput() {  
 if (estInput == null) {  
 estInput = new ArrayList<ESTInput>();  
 }  
 return this.estInput;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\est\ArrayOfESTOutput.java

package mx.mbi.dgc.client.exe.est;  
  
import java.util.ArrayList;  
import java.util.List;  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para ArrayOfESTOutput complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType name="ArrayOfESTOutput">  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="ESTOutput" type="{http://172.20.141.102/DGCest/api/estmmodel.asmx}ESTOutput" maxOccurs="unbounded" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "ArrayOfESTOutput", propOrder = {  
 "estOutput"  
})  
public class ArrayOfESTOutput {  
  
 @XmlElement(name = "ESTOutput", nillable = true)  
 protected List<ESTOutput> estOutput;  
  
 /\*\*  
 \* Gets the value of the estOutput property.  
 \*   
 \* <p>  
 \* This accessor method returns a reference to the live list,  
 \* not a snapshot. Therefore any modification you make to the  
 \* returned list will be present inside the JAXB object.  
 \* This is why there is not a <CODE>set</CODE> method for the estOutput property.  
 \*   
 \* <p>  
 \* For example, to add a new item, do as follows:  
 \* <pre>  
 \* getESTOutput().add(newItem);  
 \* </pre>  
 \*   
 \*   
 \* <p>  
 \* Objects of the following type(s) are allowed in the list  
 \* {@link ESTOutput }  
 \*   
 \*   
 \*/  
 public List<ESTOutput> getESTOutput() {  
 if (estOutput == null) {  
 estOutput = new ArrayList<ESTOutput>();  
 }  
 return this.estOutput;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\est\ESTBatch.java

package mx.mbi.dgc.client.exe.est;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para anonymous complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType>  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="inputs" type="{http://172.20.141.102/DGCest/api/estmmodel.asmx}ArrayOfESTInput" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "", propOrder = {  
 "inputs"  
})  
@XmlRootElement(name = "ESTBatch")  
public class ESTBatch {  
  
 protected ArrayOfESTInput inputs;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad inputs.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link ArrayOfESTInput }  
 \*   
 \*/  
 public ArrayOfESTInput getInputs() {  
 return inputs;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad inputs.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link ArrayOfESTInput }  
 \*   
 \*/  
 public void setInputs(ArrayOfESTInput value) {  
 this.inputs = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\est\ESTBatchResponse.java

package mx.mbi.dgc.client.exe.est;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para anonymous complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType>  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="ESTBatchResult" type="{http://172.20.141.102/DGCest/api/estmmodel.asmx}ArrayOfESTOutput" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "", propOrder = {  
 "estBatchResult"  
})  
@XmlRootElement(name = "ESTBatchResponse")  
public class ESTBatchResponse {  
  
 @XmlElement(name = "ESTBatchResult")  
 protected ArrayOfESTOutput estBatchResult;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad estBatchResult.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link ArrayOfESTOutput }  
 \*   
 \*/  
 public ArrayOfESTOutput getESTBatchResult() {  
 return estBatchResult;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad estBatchResult.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link ArrayOfESTOutput }  
 \*   
 \*/  
 public void setESTBatchResult(ArrayOfESTOutput value) {  
 this.estBatchResult = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\est\ESTInput.java

package mx.mbi.dgc.client.exe.est;  
  
import java.math.BigDecimal;  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlSeeAlso;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para ESTInput complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType name="ESTInput">  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="CurrentCapacity" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="DryBulbTemp" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="AtmosphericPressure" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="HumidityRelative" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="CalorificPower" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="SeaWaterTemp" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="PowerFactor" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "ESTInput", propOrder = {  
 "currentCapacity",  
 "dryBulbTemp",  
 "atmosphericPressure",  
 "humidityRelative",  
 "calorificPower",  
 "seaWaterTemp",  
 "powerFactor"  
})  
@XmlSeeAlso({  
 ESTOutput.class  
})  
public class ESTInput {  
  
 @XmlElement(name = "CurrentCapacity", required = true)  
 protected BigDecimal currentCapacity;  
 @XmlElement(name = "DryBulbTemp", required = true)  
 protected BigDecimal dryBulbTemp;  
 @XmlElement(name = "AtmosphericPressure", required = true)  
 protected BigDecimal atmosphericPressure;  
 @XmlElement(name = "HumidityRelative", required = true)  
 protected BigDecimal humidityRelative;  
 @XmlElement(name = "CalorificPower", required = true)  
 protected BigDecimal calorificPower;  
 @XmlElement(name = "SeaWaterTemp", required = true)  
 protected BigDecimal seaWaterTemp;  
 @XmlElement(name = "PowerFactor", required = true)  
 protected BigDecimal powerFactor;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad currentCapacity.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCurrentCapacity() {  
 return currentCapacity;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad currentCapacity.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCurrentCapacity(BigDecimal value) {  
 this.currentCapacity = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad dryBulbTemp.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getDryBulbTemp() {  
 return dryBulbTemp;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad dryBulbTemp.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setDryBulbTemp(BigDecimal value) {  
 this.dryBulbTemp = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad atmosphericPressure.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getAtmosphericPressure() {  
 return atmosphericPressure;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad atmosphericPressure.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setAtmosphericPressure(BigDecimal value) {  
 this.atmosphericPressure = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad humidityRelative.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getHumidityRelative() {  
 return humidityRelative;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad humidityRelative.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setHumidityRelative(BigDecimal value) {  
 this.humidityRelative = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad calorificPower.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCalorificPower() {  
 return calorificPower;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad calorificPower.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCalorificPower(BigDecimal value) {  
 this.calorificPower = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad seaWaterTemp.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getSeaWaterTemp() {  
 return seaWaterTemp;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad seaWaterTemp.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setSeaWaterTemp(BigDecimal value) {  
 this.seaWaterTemp = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad powerFactor.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getPowerFactor() {  
 return powerFactor;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad powerFactor.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setPowerFactor(BigDecimal value) {  
 this.powerFactor = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\est\ESTMModel.java

package mx.mbi.dgc.client.exe.est;  
  
import java.net.MalformedURLException;  
import java.net.URL;  
import javax.xml.namespace.QName;  
import javax.xml.ws.Service;  
import javax.xml.ws.WebEndpoint;  
import javax.xml.ws.WebServiceClient;  
import javax.xml.ws.WebServiceException;  
import javax.xml.ws.WebServiceFeature;  
  
  
/\*\*  
 \* This class was generated by the JAX-WS RI.  
 \* JAX-WS RI 2.2.8  
 \* Generated source version: 2.2  
 \*   
 \*/  
@WebServiceClient(name = "ESTMModel", targetNamespace = "http://172.20.141.102/DGCest/api/estmmodel.asmx", wsdlLocation = "http://172.20.141.102/DGCest/api/estmmodel.asmx?wsdl")  
public class ESTMModel  
 extends Service  
{  
  
 private final static URL ESTMMODEL\_WSDL\_LOCATION;  
 private final static WebServiceException ESTMMODEL\_EXCEPTION;  
 private final static QName ESTMMODEL\_QNAME = new QName("http://172.20.141.102/DGCest/api/estmmodel.asmx", "ESTMModel");  
  
 static {  
 URL url = null;  
 WebServiceException e = null;  
 try {  
 url = new URL("http://172.20.141.102/DGCest/api/estmmodel.asmx?wsdl");  
 } catch (MalformedURLException ex) {  
 e = new WebServiceException(ex);  
 }  
 ESTMMODEL\_WSDL\_LOCATION = url;  
 ESTMMODEL\_EXCEPTION = e;  
 }  
  
 public ESTMModel() {  
 super(\_\_getWsdlLocation(), ESTMMODEL\_QNAME);  
 }  
  
 public ESTMModel(WebServiceFeature... features) {  
 super(\_\_getWsdlLocation(), ESTMMODEL\_QNAME, features);  
 }  
  
 public ESTMModel(URL wsdlLocation) {  
 super(wsdlLocation, ESTMMODEL\_QNAME);  
 }  
  
 public ESTMModel(URL wsdlLocation, WebServiceFeature... features) {  
 super(wsdlLocation, ESTMMODEL\_QNAME, features);  
 }  
  
 public ESTMModel(URL wsdlLocation, QName serviceName) {  
 super(wsdlLocation, serviceName);  
 }  
  
 public ESTMModel(URL wsdlLocation, QName serviceName, WebServiceFeature... features) {  
 super(wsdlLocation, serviceName, features);  
 }  
  
 /\*\*  
 \*   
 \* @return  
 \* returns ESTMModelSoap  
 \*/  
 @WebEndpoint(name = "ESTMModelSoap")  
 public ESTMModelSoap getESTMModelSoap() {  
 return super.getPort(new QName("http://172.20.141.102/DGCest/api/estmmodel.asmx", "ESTMModelSoap"), ESTMModelSoap.class);  
 }  
  
 /\*\*  
 \*   
 \* @param features  
 \* A list of {@link javax.xml.ws.WebServiceFeature} to configure on the proxy. Supported features not in the <code>features</code> parameter will have their default values.  
 \* @return  
 \* returns ESTMModelSoap  
 \*/  
 @WebEndpoint(name = "ESTMModelSoap")  
 public ESTMModelSoap getESTMModelSoap(WebServiceFeature... features) {  
 return super.getPort(new QName("http://172.20.141.102/DGCest/api/estmmodel.asmx", "ESTMModelSoap"), ESTMModelSoap.class, features);  
 }  
  
 /\*\*  
 \*   
 \* @return  
 \* returns ESTMModelSoap  
 \*/  
 @WebEndpoint(name = "ESTMModelSoap12")  
 public ESTMModelSoap getESTMModelSoap12() {  
 return super.getPort(new QName("http://172.20.141.102/DGCest/api/estmmodel.asmx", "ESTMModelSoap12"), ESTMModelSoap.class);  
 }  
  
 /\*\*  
 \*   
 \* @param features  
 \* A list of {@link javax.xml.ws.WebServiceFeature} to configure on the proxy. Supported features not in the <code>features</code> parameter will have their default values.  
 \* @return  
 \* returns ESTMModelSoap  
 \*/  
 @WebEndpoint(name = "ESTMModelSoap12")  
 public ESTMModelSoap getESTMModelSoap12(WebServiceFeature... features) {  
 return super.getPort(new QName("http://172.20.141.102/DGCest/api/estmmodel.asmx", "ESTMModelSoap12"), ESTMModelSoap.class, features);  
 }  
  
 private static URL \_\_getWsdlLocation() {  
 if (ESTMMODEL\_EXCEPTION!= null) {  
 throw ESTMMODEL\_EXCEPTION;  
 }  
 return ESTMMODEL\_WSDL\_LOCATION;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\est\ESTMModelSoap.java

package mx.mbi.dgc.client.exe.est;  
  
import javax.jws.WebMethod;  
import javax.jws.WebParam;  
import javax.jws.WebResult;  
import javax.jws.WebService;  
import javax.xml.bind.annotation.XmlSeeAlso;  
import javax.xml.ws.RequestWrapper;  
import javax.xml.ws.ResponseWrapper;  
  
  
/\*\*  
 \* This class was generated by the JAX-WS RI.  
 \* JAX-WS RI 2.2.8  
 \* Generated source version: 2.2  
 \*   
 \*/  
@WebService(name = "ESTMModelSoap", targetNamespace = "http://172.20.141.102/DGCest/api/estmmodel.asmx")  
@XmlSeeAlso({  
 ObjectFactory.class  
})  
public interface ESTMModelSoap {  
  
  
 /\*\*  
 \*   
 \* @param input  
 \* @return  
 \* returns mx.mbi.dgc.client.exe.est.ESTOutput  
 \*/  
 @WebMethod(operationName = "ESTSingle", action = "http://172.20.141.102/DGCest/api/estmmodel.asmx/ESTSingle")  
 @WebResult(name = "ESTSingleResult", targetNamespace = "http://172.20.141.102/DGCest/api/estmmodel.asmx")  
 @RequestWrapper(localName = "ESTSingle", targetNamespace = "http://172.20.141.102/DGCest/api/estmmodel.asmx", className = "mx.mbi.dgc.client.exe.est.ESTSingle")  
 @ResponseWrapper(localName = "ESTSingleResponse", targetNamespace = "http://172.20.141.102/DGCest/api/estmmodel.asmx", className = "mx.mbi.dgc.client.exe.est.ESTSingleResponse")  
 public ESTOutput estSingle(  
 @WebParam(name = "input", targetNamespace = "http://172.20.141.102/DGCest/api/estmmodel.asmx")  
 ESTInput input);  
  
 /\*\*  
 \*   
 \* @param inputs  
 \* @return  
 \* returns mx.mbi.dgc.client.exe.est.ArrayOfESTOutput  
 \*/  
 @WebMethod(operationName = "ESTBatch", action = "http://172.20.141.102/DGCest/api/estmmodel.asmx/ESTBatch")  
 @WebResult(name = "ESTBatchResult", targetNamespace = "http://172.20.141.102/DGCest/api/estmmodel.asmx")  
 @RequestWrapper(localName = "ESTBatch", targetNamespace = "http://172.20.141.102/DGCest/api/estmmodel.asmx", className = "mx.mbi.dgc.client.exe.est.ESTBatch")  
 @ResponseWrapper(localName = "ESTBatchResponse", targetNamespace = "http://172.20.141.102/DGCest/api/estmmodel.asmx", className = "mx.mbi.dgc.client.exe.est.ESTBatchResponse")  
 public ArrayOfESTOutput estBatch(  
 @WebParam(name = "inputs", targetNamespace = "http://172.20.141.102/DGCest/api/estmmodel.asmx")  
 ArrayOfESTInput inputs);  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\est\ESTOutput.java

package mx.mbi.dgc.client.exe.est;  
  
import java.math.BigDecimal;  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para ESTOutput complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType name="ESTOutput">  
 \* &lt;complexContent>  
 \* &lt;extension base="{http://172.20.141.102/DGCest/api/estmmodel.asmx}ESTInput">  
 \* &lt;sequence>  
 \* &lt;element name="CurrentPlantLoad" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="NetCapacity" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="SummerCapacity" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="CTUNGLoad" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>  
 \* &lt;element name="CTUNG" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>  
 \* &lt;element name="CTOV" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>  
 \* &lt;element name="Cpw\_AT" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Cpw\_BP" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Cpw\_RH" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Cpw\_PCI" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Cpw\_CW" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Cpw\_FP" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Chr\_AT" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Chr\_BP" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Chr\_RH" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Chr\_PCI" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Chr\_CW" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;element name="Chr\_FP" type="{http://www.w3.org/2001/XMLSchema}decimal"/>  
 \* &lt;/sequence>  
 \* &lt;/extension>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "ESTOutput", propOrder = {  
 "currentPlantLoad",  
 "netCapacity",  
 "summerCapacity",  
 "ctungLoad",  
 "ctung",  
 "ctov",  
 "cpwAT",  
 "cpwBP",  
 "cpwRH",  
 "cpwPCI",  
 "cpwCW",  
 "cpwFP",  
 "chrAT",  
 "chrBP",  
 "chrRH",  
 "chrPCI",  
 "chrCW",  
 "chrFP"  
})  
public class ESTOutput  
 extends ESTInput  
{  
  
 @XmlElement(name = "CurrentPlantLoad", required = true)  
 protected BigDecimal currentPlantLoad;  
 @XmlElement(name = "NetCapacity", required = true)  
 protected BigDecimal netCapacity;  
 @XmlElement(name = "SummerCapacity", required = true)  
 protected BigDecimal summerCapacity;  
 @XmlElement(name = "CTUNGLoad")  
 protected String ctungLoad;  
 @XmlElement(name = "CTUNG")  
 protected String ctung;  
 @XmlElement(name = "CTOV")  
 protected String ctov;  
 @XmlElement(name = "Cpw\_AT", required = true)  
 protected BigDecimal cpwAT;  
 @XmlElement(name = "Cpw\_BP", required = true)  
 protected BigDecimal cpwBP;  
 @XmlElement(name = "Cpw\_RH", required = true)  
 protected BigDecimal cpwRH;  
 @XmlElement(name = "Cpw\_PCI", required = true)  
 protected BigDecimal cpwPCI;  
 @XmlElement(name = "Cpw\_CW", required = true)  
 protected BigDecimal cpwCW;  
 @XmlElement(name = "Cpw\_FP", required = true)  
 protected BigDecimal cpwFP;  
 @XmlElement(name = "Chr\_AT", required = true)  
 protected BigDecimal chrAT;  
 @XmlElement(name = "Chr\_BP", required = true)  
 protected BigDecimal chrBP;  
 @XmlElement(name = "Chr\_RH", required = true)  
 protected BigDecimal chrRH;  
 @XmlElement(name = "Chr\_PCI", required = true)  
 protected BigDecimal chrPCI;  
 @XmlElement(name = "Chr\_CW", required = true)  
 protected BigDecimal chrCW;  
 @XmlElement(name = "Chr\_FP", required = true)  
 protected BigDecimal chrFP;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad currentPlantLoad.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCurrentPlantLoad() {  
 return currentPlantLoad;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad currentPlantLoad.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCurrentPlantLoad(BigDecimal value) {  
 this.currentPlantLoad = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad netCapacity.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getNetCapacity() {  
 return netCapacity;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad netCapacity.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setNetCapacity(BigDecimal value) {  
 this.netCapacity = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad summerCapacity.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getSummerCapacity() {  
 return summerCapacity;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad summerCapacity.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setSummerCapacity(BigDecimal value) {  
 this.summerCapacity = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad ctungLoad.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getCTUNGLoad() {  
 return ctungLoad;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad ctungLoad.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setCTUNGLoad(String value) {  
 this.ctungLoad = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad ctung.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getCTUNG() {  
 return ctung;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad ctung.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setCTUNG(String value) {  
 this.ctung = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad ctov.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getCTOV() {  
 return ctov;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad ctov.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setCTOV(String value) {  
 this.ctov = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad cpwAT.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCpwAT() {  
 return cpwAT;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad cpwAT.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCpwAT(BigDecimal value) {  
 this.cpwAT = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad cpwBP.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCpwBP() {  
 return cpwBP;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad cpwBP.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCpwBP(BigDecimal value) {  
 this.cpwBP = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad cpwRH.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCpwRH() {  
 return cpwRH;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad cpwRH.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCpwRH(BigDecimal value) {  
 this.cpwRH = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad cpwPCI.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCpwPCI() {  
 return cpwPCI;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad cpwPCI.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCpwPCI(BigDecimal value) {  
 this.cpwPCI = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad cpwCW.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCpwCW() {  
 return cpwCW;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad cpwCW.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCpwCW(BigDecimal value) {  
 this.cpwCW = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad cpwFP.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getCpwFP() {  
 return cpwFP;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad cpwFP.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setCpwFP(BigDecimal value) {  
 this.cpwFP = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad chrAT.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getChrAT() {  
 return chrAT;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad chrAT.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setChrAT(BigDecimal value) {  
 this.chrAT = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad chrBP.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getChrBP() {  
 return chrBP;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad chrBP.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setChrBP(BigDecimal value) {  
 this.chrBP = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad chrRH.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getChrRH() {  
 return chrRH;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad chrRH.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setChrRH(BigDecimal value) {  
 this.chrRH = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad chrPCI.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getChrPCI() {  
 return chrPCI;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad chrPCI.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setChrPCI(BigDecimal value) {  
 this.chrPCI = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad chrCW.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getChrCW() {  
 return chrCW;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad chrCW.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setChrCW(BigDecimal value) {  
 this.chrCW = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad chrFP.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public BigDecimal getChrFP() {  
 return chrFP;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad chrFP.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link BigDecimal }  
 \*   
 \*/  
 public void setChrFP(BigDecimal value) {  
 this.chrFP = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\est\ESTSingle.java

package mx.mbi.dgc.client.exe.est;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para anonymous complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType>  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="input" type="{http://172.20.141.102/DGCest/api/estmmodel.asmx}ESTInput" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "", propOrder = {  
 "input"  
})  
@XmlRootElement(name = "ESTSingle")  
public class ESTSingle {  
  
 protected ESTInput input;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad input.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link ESTInput }  
 \*   
 \*/  
 public ESTInput getInput() {  
 return input;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad input.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link ESTInput }  
 \*   
 \*/  
 public void setInput(ESTInput value) {  
 this.input = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\est\ESTSingleResponse.java

package mx.mbi.dgc.client.exe.est;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para anonymous complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType>  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="ESTSingleResult" type="{http://172.20.141.102/DGCest/api/estmmodel.asmx}ESTOutput" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "", propOrder = {  
 "estSingleResult"  
})  
@XmlRootElement(name = "ESTSingleResponse")  
public class ESTSingleResponse {  
  
 @XmlElement(name = "ESTSingleResult")  
 protected ESTOutput estSingleResult;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad estSingleResult.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link ESTOutput }  
 \*   
 \*/  
 public ESTOutput getESTSingleResult() {  
 return estSingleResult;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad estSingleResult.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link ESTOutput }  
 \*   
 \*/  
 public void setESTSingleResult(ESTOutput value) {  
 this.estSingleResult = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\est\ObjectFactory.java

package mx.mbi.dgc.client.exe.est;  
  
import javax.xml.bind.annotation.XmlRegistry;  
  
  
/\*\*  
 \* This object contains factory methods for each   
 \* Java content interface and Java element interface   
 \* generated in the mx.mbi.dgc.client.exe.est package.   
 \* <p>An ObjectFactory allows you to programatically   
 \* construct new instances of the Java representation   
 \* for XML content. The Java representation of XML   
 \* content can consist of schema derived interfaces   
 \* and classes representing the binding of schema   
 \* type definitions, element declarations and model   
 \* groups. Factory methods for each of these are   
 \* provided in this class.  
 \*   
 \*/  
@XmlRegistry  
public class ObjectFactory {  
  
  
 /\*\*  
 \* Create a new ObjectFactory that can be used to create new instances of schema derived classes for package: mx.mbi.dgc.client.exe.est  
 \*   
 \*/  
 public ObjectFactory() {  
 }  
  
 /\*\*  
 \* Create an instance of {@link ESTSingleResponse }  
 \*   
 \*/  
 public ESTSingleResponse createESTSingleResponse() {  
 return new ESTSingleResponse();  
 }  
  
 /\*\*  
 \* Create an instance of {@link ESTOutput }  
 \*   
 \*/  
 public ESTOutput createESTOutput() {  
 return new ESTOutput();  
 }  
  
 /\*\*  
 \* Create an instance of {@link ESTSingle }  
 \*   
 \*/  
 public ESTSingle createESTSingle() {  
 return new ESTSingle();  
 }  
  
 /\*\*  
 \* Create an instance of {@link ESTInput }  
 \*   
 \*/  
 public ESTInput createESTInput() {  
 return new ESTInput();  
 }  
  
 /\*\*  
 \* Create an instance of {@link ESTBatch }  
 \*   
 \*/  
 public ESTBatch createESTBatch() {  
 return new ESTBatch();  
 }  
  
 /\*\*  
 \* Create an instance of {@link ArrayOfESTInput }  
 \*   
 \*/  
 public ArrayOfESTInput createArrayOfESTInput() {  
 return new ArrayOfESTInput();  
 }  
  
 /\*\*  
 \* Create an instance of {@link ESTBatchResponse }  
 \*   
 \*/  
 public ESTBatchResponse createESTBatchResponse() {  
 return new ESTBatchResponse();  
 }  
  
 /\*\*  
 \* Create an instance of {@link ArrayOfESTOutput }  
 \*   
 \*/  
 public ArrayOfESTOutput createArrayOfESTOutput() {  
 return new ArrayOfESTOutput();  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\exe\est\package-info.java

@javax.xml.bind.annotation.XmlSchema(namespace = "http://172.20.141.102/DGCest/api/estmmodel.asmx", elementFormDefault = javax.xml.bind.annotation.XmlNsForm.QUALIFIED)  
package mx.mbi.dgc.client.exe.est;

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\facturacion\ArrayOfVerificationError.java

package mx.mbi.dgc.client.facturacion;  
  
import java.util.ArrayList;  
import java.util.List;  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para ArrayOfVerificationError complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType name="ArrayOfVerificationError">  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="VerificationError" type="{https://cfdi.timbrado.com.mx/timbradov2}VerificationError" maxOccurs="unbounded" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "ArrayOfVerificationError", propOrder = {  
 "verificationError"  
})  
public class ArrayOfVerificationError {  
  
 @XmlElement(name = "VerificationError")  
 protected List<VerificationError> verificationError;  
  
 /\*\*  
 \* Gets the value of the verificationError property.  
 \*   
 \* <p>  
 \* This accessor method returns a reference to the live list,  
 \* not a snapshot. Therefore any modification you make to the  
 \* returned list will be present inside the JAXB object.  
 \* This is why there is not a <CODE>set</CODE> method for the verificationError property.  
 \*   
 \* <p>  
 \* For example, to add a new item, do as follows:  
 \* <pre>  
 \* getVerificationError().add(newItem);  
 \* </pre>  
 \*   
 \*   
 \* <p>  
 \* Objects of the following type(s) are allowed in the list  
 \* {@link VerificationError }  
 \*   
 \*   
 \*/  
 public List<VerificationError> getVerificationError() {  
 if (verificationError == null) {  
 verificationError = new ArrayList<VerificationError>();  
 }  
 return this.verificationError;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\facturacion\AuthenticationHeader.java

package mx.mbi.dgc.client.facturacion;  
  
import java.util.HashMap;  
import java.util.Map;  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlAnyAttribute;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlType;  
import javax.xml.namespace.QName;  
  
  
/\*\*  
 \* <p>Clase Java para AuthenticationHeader complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType name="AuthenticationHeader">  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="UserName" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>  
 \* &lt;element name="Password" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;anyAttribute/>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "AuthenticationHeader", propOrder = {  
 "userName",  
 "password"  
})  
public class AuthenticationHeader {  
  
 @XmlElement(name = "UserName")  
 protected String userName;  
 @XmlElement(name = "Password")  
 protected String password;  
 @XmlAnyAttribute  
 private Map<QName, String> otherAttributes = new HashMap<QName, String>();  
  
 /\*\*  
 \* Obtiene el valor de la propiedad userName.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getUserName() {  
 return userName;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad userName.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setUserName(String value) {  
 this.userName = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad password.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getPassword() {  
 return password;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad password.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setPassword(String value) {  
 this.password = value;  
 }  
  
 /\*\*  
 \* Gets a map that contains attributes that aren't bound to any typed property on this class.  
 \*   
 \* <p>  
 \* the map is keyed by the name of the attribute and   
 \* the value is the string value of the attribute.  
 \*   
 \* the map returned by this method is live, and you can add new attribute  
 \* by updating the map directly. Because of this design, there's no setter.  
 \*   
 \*   
 \* @return  
 \* always non-null  
 \*/  
 public Map<QName, String> getOtherAttributes() {  
 return otherAttributes;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\facturacion\Cfdi.java

package mx.mbi.dgc.client.facturacion;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlAttribute;  
import javax.xml.bind.annotation.XmlSchemaType;  
import javax.xml.bind.annotation.XmlType;  
import javax.xml.datatype.XMLGregorianCalendar;  
  
  
/\*\*  
 \* <p>Clase Java para cfdi complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType name="cfdi">  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;attribute name="Id" type="{http://www.w3.org/2001/XMLSchema}string" />  
 \* &lt;attribute name="FechaHora" use="required" type="{http://www.w3.org/2001/XMLSchema}dateTime" />  
 \* &lt;attribute name="RFCEmisor" type="{http://www.w3.org/2001/XMLSchema}string" />  
 \* &lt;attribute name="RFCReceptor" type="{http://www.w3.org/2001/XMLSchema}string" />  
 \* &lt;attribute name="Serie" type="{http://www.w3.org/2001/XMLSchema}string" />  
 \* &lt;attribute name="Folio" type="{http://www.w3.org/2001/XMLSchema}string" />  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "cfdi")  
public class Cfdi {  
  
 @XmlAttribute(name = "Id")  
 protected String id;  
 @XmlAttribute(name = "FechaHora", required = true)  
 @XmlSchemaType(name = "dateTime")  
 protected XMLGregorianCalendar fechaHora;  
 @XmlAttribute(name = "RFCEmisor")  
 protected String rfcEmisor;  
 @XmlAttribute(name = "RFCReceptor")  
 protected String rfcReceptor;  
 @XmlAttribute(name = "Serie")  
 protected String serie;  
 @XmlAttribute(name = "Folio")  
 protected String folio;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad id.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getId() {  
 return id;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad id.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setId(String value) {  
 this.id = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad fechaHora.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link XMLGregorianCalendar }  
 \*   
 \*/  
 public XMLGregorianCalendar getFechaHora() {  
 return fechaHora;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad fechaHora.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link XMLGregorianCalendar }  
 \*   
 \*/  
 public void setFechaHora(XMLGregorianCalendar value) {  
 this.fechaHora = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad rfcEmisor.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getRFCEmisor() {  
 return rfcEmisor;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad rfcEmisor.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setRFCEmisor(String value) {  
 this.rfcEmisor = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad rfcReceptor.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getRFCReceptor() {  
 return rfcReceptor;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad rfcReceptor.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setRFCReceptor(String value) {  
 this.rfcReceptor = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad serie.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getSerie() {  
 return serie;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad serie.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setSerie(String value) {  
 this.serie = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad folio.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getFolio() {  
 return folio;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad folio.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setFolio(String value) {  
 this.folio = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\facturacion\GeneraTimbre.java

package mx.mbi.dgc.client.facturacion;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para anonymous complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType>  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="cfdiBytes" type="{http://www.w3.org/2001/XMLSchema}base64Binary" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "", propOrder = {  
 "cfdiBytes"  
})  
@XmlRootElement(name = "GeneraTimbre")  
public class GeneraTimbre {  
  
 protected byte[] cfdiBytes;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad cfdiBytes.  
 \*   
 \* @return  
 \* possible object is  
 \* byte[]  
 \*/  
 public byte[] getCfdiBytes() {  
 return cfdiBytes;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad cfdiBytes.  
 \*   
 \* @param value  
 \* allowed object is  
 \* byte[]  
 \*/  
 public void setCfdiBytes(byte[] value) {  
 this.cfdiBytes = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\facturacion\GeneraTimbreResponse.java

package mx.mbi.dgc.client.facturacion;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para anonymous complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType>  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="GeneraTimbreResult" type="{https://cfdi.timbrado.com.mx/timbradov2}Response" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "", propOrder = {  
 "generaTimbreResult"  
})  
@XmlRootElement(name = "GeneraTimbreResponse")  
public class GeneraTimbreResponse {  
  
 @XmlElement(name = "GeneraTimbreResult")  
 protected Response generaTimbreResult;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad generaTimbreResult.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link Response }  
 \*   
 \*/  
 public Response getGeneraTimbreResult() {  
 return generaTimbreResult;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad generaTimbreResult.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link Response }  
 \*   
 \*/  
 public void setGeneraTimbreResult(Response value) {  
 this.generaTimbreResult = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\facturacion\ObjectFactory.java

package mx.mbi.dgc.client.facturacion;  
  
import javax.xml.bind.JAXBElement;  
import javax.xml.bind.annotation.XmlElementDecl;  
import javax.xml.bind.annotation.XmlRegistry;  
import javax.xml.namespace.QName;  
  
  
/\*\*  
 \* This object contains factory methods for each   
 \* Java content interface and Java element interface   
 \* generated in the mx.mbi.dgc.client.facturacion package.   
 \* <p>An ObjectFactory allows you to programatically   
 \* construct new instances of the Java representation   
 \* for XML content. The Java representation of XML   
 \* content can consist of schema derived interfaces   
 \* and classes representing the binding of schema   
 \* type definitions, element declarations and model   
 \* groups. Factory methods for each of these are   
 \* provided in this class.  
 \*   
 \*/  
@XmlRegistry  
public class ObjectFactory {  
  
 private final static QName \_AuthenticationHeader\_QNAME = new QName("https://cfdi.timbrado.com.mx/timbradov2", "AuthenticationHeader");  
 private final static QName \_ResponseHeader\_QNAME = new QName("https://cfdi.timbrado.com.mx/timbradov2", "ResponseHeader");  
 private final static QName \_TestConfigHeader\_QNAME = new QName("https://cfdi.timbrado.com.mx/timbradov2", "TestConfigHeader");  
  
 /\*\*  
 \* Create a new ObjectFactory that can be used to create new instances of schema derived classes for package: mx.mbi.dgc.client.facturacion  
 \*   
 \*/  
 public ObjectFactory() {  
 }  
  
 /\*\*  
 \* Create an instance of {@link GeneraTimbre }  
 \*   
 \*/  
 public GeneraTimbre createGeneraTimbre() {  
 return new GeneraTimbre();  
 }  
  
 /\*\*  
 \* Create an instance of {@link GeneraTimbreResponse }  
 \*   
 \*/  
 public GeneraTimbreResponse createGeneraTimbreResponse() {  
 return new GeneraTimbreResponse();  
 }  
  
 /\*\*  
 \* Create an instance of {@link Response }  
 \*   
 \*/  
 public Response createResponse() {  
 return new Response();  
 }  
  
 /\*\*  
 \* Create an instance of {@link ResponseHeader }  
 \*   
 \*/  
 public ResponseHeader createResponseHeader() {  
 return new ResponseHeader();  
 }  
  
 /\*\*  
 \* Create an instance of {@link AuthenticationHeader }  
 \*   
 \*/  
 public AuthenticationHeader createAuthenticationHeader() {  
 return new AuthenticationHeader();  
 }  
  
 /\*\*  
 \* Create an instance of {@link TestConfigHeader }  
 \*   
 \*/  
 public TestConfigHeader createTestConfigHeader() {  
 return new TestConfigHeader();  
 }  
  
 /\*\*  
 \* Create an instance of {@link ArrayOfVerificationError }  
 \*   
 \*/  
 public ArrayOfVerificationError createArrayOfVerificationError() {  
 return new ArrayOfVerificationError();  
 }  
  
 /\*\*  
 \* Create an instance of {@link VerificationError }  
 \*   
 \*/  
 public VerificationError createVerificationError() {  
 return new VerificationError();  
 }  
  
 /\*\*  
 \* Create an instance of {@link ResponseError }  
 \*   
 \*/  
 public ResponseError createResponseError() {  
 return new ResponseError();  
 }  
  
 /\*\*  
 \* Create an instance of {@link Cfdi }  
 \*   
 \*/  
 public Cfdi createCfdi() {  
 return new Cfdi();  
 }  
  
 /\*\*  
 \* Create an instance of {@link JAXBElement }{@code <}{@link AuthenticationHeader }{@code >}}  
 \*   
 \*/  
 @XmlElementDecl(namespace = "https://cfdi.timbrado.com.mx/timbradov2", name = "AuthenticationHeader")  
 public JAXBElement<AuthenticationHeader> createAuthenticationHeader(AuthenticationHeader value) {  
 return new JAXBElement<AuthenticationHeader>(\_AuthenticationHeader\_QNAME, AuthenticationHeader.class, null, value);  
 }  
  
 /\*\*  
 \* Create an instance of {@link JAXBElement }{@code <}{@link ResponseHeader }{@code >}}  
 \*   
 \*/  
 @XmlElementDecl(namespace = "https://cfdi.timbrado.com.mx/timbradov2", name = "ResponseHeader")  
 public JAXBElement<ResponseHeader> createResponseHeader(ResponseHeader value) {  
 return new JAXBElement<ResponseHeader>(\_ResponseHeader\_QNAME, ResponseHeader.class, null, value);  
 }  
  
 /\*\*  
 \* Create an instance of {@link JAXBElement }{@code <}{@link TestConfigHeader }{@code >}}  
 \*   
 \*/  
 @XmlElementDecl(namespace = "https://cfdi.timbrado.com.mx/timbradov2", name = "TestConfigHeader")  
 public JAXBElement<TestConfigHeader> createTestConfigHeader(TestConfigHeader value) {  
 return new JAXBElement<TestConfigHeader>(\_TestConfigHeader\_QNAME, TestConfigHeader.class, null, value);  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\facturacion\package-info.java

@javax.xml.bind.annotation.XmlSchema(namespace = "https://cfdi.timbrado.com.mx/timbradov2", elementFormDefault = javax.xml.bind.annotation.XmlNsForm.QUALIFIED)  
package mx.mbi.dgc.client.facturacion;

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\facturacion\Response.java

package mx.mbi.dgc.client.facturacion;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para Response complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType name="Response">  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="Timbre" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>  
 \* &lt;element name="Error" type="{https://cfdi.timbrado.com.mx/timbradov2}ResponseError" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "Response", propOrder = {  
 "timbre",  
 "error"  
})  
public class Response {  
  
 @XmlElement(name = "Timbre")  
 protected String timbre;  
 @XmlElement(name = "Error")  
 protected ResponseError error;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad timbre.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getTimbre() {  
 return timbre;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad timbre.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setTimbre(String value) {  
 this.timbre = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad error.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link ResponseError }  
 \*   
 \*/  
 public ResponseError getError() {  
 return error;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad error.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link ResponseError }  
 \*   
 \*/  
 public void setError(ResponseError value) {  
 this.error = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\facturacion\ResponseError.java

package mx.mbi.dgc.client.facturacion;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlAttribute;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para ResponseError complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType name="ResponseError">  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="ErroresVerificacion" type="{https://cfdi.timbrado.com.mx/timbradov2}ArrayOfVerificationError" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;attribute name="Codigo" type="{http://www.w3.org/2001/XMLSchema}string" />  
 \* &lt;attribute name="Descripcion" type="{http://www.w3.org/2001/XMLSchema}string" />  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "ResponseError", propOrder = {  
 "erroresVerificacion"  
})  
public class ResponseError {  
  
 @XmlElement(name = "ErroresVerificacion")  
 protected ArrayOfVerificationError erroresVerificacion;  
 @XmlAttribute(name = "Codigo")  
 protected String codigo;  
 @XmlAttribute(name = "Descripcion")  
 protected String descripcion;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad erroresVerificacion.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link ArrayOfVerificationError }  
 \*   
 \*/  
 public ArrayOfVerificationError getErroresVerificacion() {  
 return erroresVerificacion;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad erroresVerificacion.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link ArrayOfVerificationError }  
 \*   
 \*/  
 public void setErroresVerificacion(ArrayOfVerificationError value) {  
 this.erroresVerificacion = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad codigo.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getCodigo() {  
 return codigo;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad codigo.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setCodigo(String value) {  
 this.codigo = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad descripcion.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getDescripcion() {  
 return descripcion;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad descripcion.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setDescripcion(String value) {  
 this.descripcion = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\facturacion\ResponseHeader.java

package mx.mbi.dgc.client.facturacion;  
  
import java.util.HashMap;  
import java.util.Map;  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlAnyAttribute;  
import javax.xml.bind.annotation.XmlAttribute;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlSchemaType;  
import javax.xml.bind.annotation.XmlType;  
import javax.xml.datatype.XMLGregorianCalendar;  
import javax.xml.namespace.QName;  
  
  
/\*\*  
 \* <p>Clase Java para ResponseHeader complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType name="ResponseHeader">  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="CFDI" type="{https://cfdi.timbrado.com.mx/timbradov2}cfdi" minOccurs="0"/>  
 \* &lt;/sequence>  
 \* &lt;attribute name="trackId" use="required" type="{http://www.w3.org/2001/XMLSchema}long" />  
 \* &lt;attribute name="RequestDateTime" use="required" type="{http://www.w3.org/2001/XMLSchema}dateTime" />  
 \* &lt;attribute name="IPCaller" type="{http://www.w3.org/2001/XMLSchema}string" />  
 \* &lt;attribute name="ClienteId" use="required" type="{http://www.w3.org/2001/XMLSchema}int" />  
 \* &lt;attribute name="Estatus" use="required" type="{http://www.w3.org/2001/XMLSchema}int" />  
 \* &lt;attribute name="ID" type="{http://www.w3.org/2001/XMLSchema}string" />  
 \* &lt;attribute name="Detail" type="{http://www.w3.org/2001/XMLSchema}string" />  
 \* &lt;attribute name="Ambiente" type="{http://www.w3.org/2001/XMLSchema}string" />  
 \* &lt;anyAttribute/>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "ResponseHeader", propOrder = {  
 "cfdi"  
})  
public class ResponseHeader {  
  
 @XmlElement(name = "CFDI")  
 protected Cfdi cfdi;  
 @XmlAttribute(name = "trackId", required = true)  
 protected long trackId;  
 @XmlAttribute(name = "RequestDateTime", required = true)  
 @XmlSchemaType(name = "dateTime")  
 protected XMLGregorianCalendar requestDateTime;  
 @XmlAttribute(name = "IPCaller")  
 protected String ipCaller;  
 @XmlAttribute(name = "ClienteId", required = true)  
 protected int clienteId;  
 @XmlAttribute(name = "Estatus", required = true)  
 protected int estatus;  
 @XmlAttribute(name = "ID")  
 protected String id;  
 @XmlAttribute(name = "Detail")  
 protected String detail;  
 @XmlAttribute(name = "Ambiente")  
 protected String ambiente;  
 @XmlAnyAttribute  
 private Map<QName, String> otherAttributes = new HashMap<QName, String>();  
  
 /\*\*  
 \* Obtiene el valor de la propiedad cfdi.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link Cfdi }  
 \*   
 \*/  
 public Cfdi getCFDI() {  
 return cfdi;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad cfdi.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link Cfdi }  
 \*   
 \*/  
 public void setCFDI(Cfdi value) {  
 this.cfdi = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad trackId.  
 \*   
 \*/  
 public long getTrackId() {  
 return trackId;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad trackId.  
 \*   
 \*/  
 public void setTrackId(long value) {  
 this.trackId = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad requestDateTime.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link XMLGregorianCalendar }  
 \*   
 \*/  
 public XMLGregorianCalendar getRequestDateTime() {  
 return requestDateTime;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad requestDateTime.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link XMLGregorianCalendar }  
 \*   
 \*/  
 public void setRequestDateTime(XMLGregorianCalendar value) {  
 this.requestDateTime = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad ipCaller.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getIPCaller() {  
 return ipCaller;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad ipCaller.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setIPCaller(String value) {  
 this.ipCaller = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad clienteId.  
 \*   
 \*/  
 public int getClienteId() {  
 return clienteId;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad clienteId.  
 \*   
 \*/  
 public void setClienteId(int value) {  
 this.clienteId = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad estatus.  
 \*   
 \*/  
 public int getEstatus() {  
 return estatus;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad estatus.  
 \*   
 \*/  
 public void setEstatus(int value) {  
 this.estatus = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad id.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getID() {  
 return id;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad id.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setID(String value) {  
 this.id = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad detail.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getDetail() {  
 return detail;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad detail.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setDetail(String value) {  
 this.detail = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad ambiente.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getAmbiente() {  
 return ambiente;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad ambiente.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setAmbiente(String value) {  
 this.ambiente = value;  
 }  
  
 /\*\*  
 \* Gets a map that contains attributes that aren't bound to any typed property on this class.  
 \*   
 \* <p>  
 \* the map is keyed by the name of the attribute and   
 \* the value is the string value of the attribute.  
 \*   
 \* the map returned by this method is live, and you can add new attribute  
 \* by updating the map directly. Because of this design, there's no setter.  
 \*   
 \*   
 \* @return  
 \* always non-null  
 \*/  
 public Map<QName, String> getOtherAttributes() {  
 return otherAttributes;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\facturacion\TestConfigHeader.java

package mx.mbi.dgc.client.facturacion;  
  
import java.util.HashMap;  
import java.util.Map;  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlAnyAttribute;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlType;  
import javax.xml.namespace.QName;  
  
  
/\*\*  
 \* <p>Clase Java para TestConfigHeader complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType name="TestConfigHeader">  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;sequence>  
 \* &lt;element name="ConsultaDB" type="{http://www.w3.org/2001/XMLSchema}boolean"/>  
 \* &lt;element name="UUIDPrueba" type="{http://www.w3.org/2001/XMLSchema}boolean"/>  
 \* &lt;element name="TimbrarDuplicados" type="{http://www.w3.org/2001/XMLSchema}boolean"/>  
 \* &lt;/sequence>  
 \* &lt;anyAttribute/>  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "TestConfigHeader", propOrder = {  
 "consultaDB",  
 "uuidPrueba",  
 "timbrarDuplicados"  
})  
public class TestConfigHeader {  
  
 @XmlElement(name = "ConsultaDB")  
 protected boolean consultaDB;  
 @XmlElement(name = "UUIDPrueba")  
 protected boolean uuidPrueba;  
 @XmlElement(name = "TimbrarDuplicados")  
 protected boolean timbrarDuplicados;  
 @XmlAnyAttribute  
 private Map<QName, String> otherAttributes = new HashMap<QName, String>();  
  
 /\*\*  
 \* Obtiene el valor de la propiedad consultaDB.  
 \*   
 \*/  
 public boolean isConsultaDB() {  
 return consultaDB;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad consultaDB.  
 \*   
 \*/  
 public void setConsultaDB(boolean value) {  
 this.consultaDB = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad uuidPrueba.  
 \*   
 \*/  
 public boolean isUUIDPrueba() {  
 return uuidPrueba;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad uuidPrueba.  
 \*   
 \*/  
 public void setUUIDPrueba(boolean value) {  
 this.uuidPrueba = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad timbrarDuplicados.  
 \*   
 \*/  
 public boolean isTimbrarDuplicados() {  
 return timbrarDuplicados;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad timbrarDuplicados.  
 \*   
 \*/  
 public void setTimbrarDuplicados(boolean value) {  
 this.timbrarDuplicados = value;  
 }  
  
 /\*\*  
 \* Gets a map that contains attributes that aren't bound to any typed property on this class.  
 \*   
 \* <p>  
 \* the map is keyed by the name of the attribute and   
 \* the value is the string value of the attribute.  
 \*   
 \* the map returned by this method is live, and you can add new attribute  
 \* by updating the map directly. Because of this design, there's no setter.  
 \*   
 \*   
 \* @return  
 \* always non-null  
 \*/  
 public Map<QName, String> getOtherAttributes() {  
 return otherAttributes;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\facturacion\VerificationError.java

package mx.mbi.dgc.client.facturacion;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlAttribute;  
import javax.xml.bind.annotation.XmlType;  
  
  
/\*\*  
 \* <p>Clase Java para VerificationError complex type.  
 \*   
 \* <p>El siguiente fragmento de esquema especifica el contenido que se espera que haya en esta clase.  
 \*   
 \* <pre>  
 \* &lt;complexType name="VerificationError">  
 \* &lt;complexContent>  
 \* &lt;restriction base="{http://www.w3.org/2001/XMLSchema}anyType">  
 \* &lt;attribute name="Codigo" type="{http://www.w3.org/2001/XMLSchema}string" />  
 \* &lt;attribute name="Descripcion" type="{http://www.w3.org/2001/XMLSchema}string" />  
 \* &lt;/restriction>  
 \* &lt;/complexContent>  
 \* &lt;/complexType>  
 \* </pre>  
 \*   
 \*   
 \*/  
@XmlAccessorType(XmlAccessType.FIELD)  
@XmlType(name = "VerificationError")  
public class VerificationError {  
  
 @XmlAttribute(name = "Codigo")  
 protected String codigo;  
 @XmlAttribute(name = "Descripcion")  
 protected String descripcion;  
  
 /\*\*  
 \* Obtiene el valor de la propiedad codigo.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getCodigo() {  
 return codigo;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad codigo.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setCodigo(String value) {  
 this.codigo = value;  
 }  
  
 /\*\*  
 \* Obtiene el valor de la propiedad descripcion.  
 \*   
 \* @return  
 \* possible object is  
 \* {@link String }  
 \*   
 \*/  
 public String getDescripcion() {  
 return descripcion;  
 }  
  
 /\*\*  
 \* Define el valor de la propiedad descripcion.  
 \*   
 \* @param value  
 \* allowed object is  
 \* {@link String }  
 \*   
 \*/  
 public void setDescripcion(String value) {  
 this.descripcion = value;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\facturacion\WsTimbrado.java

package mx.mbi.dgc.client.facturacion;  
  
  
import java.net.MalformedURLException;  
import java.net.URL;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
import java.util.Map;  
import java.util.Set;  
  
import javax.xml.bind.JAXBContext;  
import javax.xml.bind.JAXBElement;  
import javax.xml.bind.Marshaller;  
import javax.xml.namespace.QName;  
import javax.xml.soap.SOAPHeader;  
import javax.xml.ws.Binding;  
import javax.xml.ws.BindingProvider;  
import javax.xml.ws.Service;  
import javax.xml.ws.WebEndpoint;  
import javax.xml.ws.WebServiceClient;  
import javax.xml.ws.WebServiceException;  
import javax.xml.ws.WebServiceFeature;  
import javax.xml.ws.handler.Handler;  
import javax.xml.ws.handler.MessageContext;  
import javax.xml.ws.handler.soap.SOAPHandler;  
import javax.xml.ws.handler.soap.SOAPMessageContext;  
  
/\*\*  
 \* This class was generated by the JAX-WS RI.  
 \* JAX-WS RI 2.2.6-1b01  
 \* Generated source version: 2.2  
 \*  
 \*/  
@WebServiceClient(name = "wsTimbrado", targetNamespace = "https://cfdi.timbrado.com.mx/timbradov2",  
 wsdlLocation = "http://201.144.64.67:81/wsTimbrado.asmx?wsdl")  
public class WsTimbrado extends Service {  
  
 private final static URL WSTIMBRADO\_WSDL\_LOCATION;   
 private final static WebServiceException WSTIMBRADO\_EXCEPTION;  
 private final static QName WSTIMBRADO\_QNAME = new QName("https://cfdi.timbrado.com.mx/timbradov2", "wsTimbrado");  
  
 static {  
 URL url = null;  
 WebServiceException e = null;  
  
 ObjectFactory factory = new ObjectFactory();  
 AuthenticationHeader value = new AuthenticationHeader();  
 value.setUserName("autofac\_t");  
 value.setPassword("6Pj!N+5sbQ$4=t8Y");  
 factory.createAuthenticationHeader(value);  
  
 try {  
 url = new URL("http://201.144.64.67:81/wsTimbrado.asmx?wsdl");  
 } catch (MalformedURLException ex) {  
 ex.printStackTrace();  
 e = new WebServiceException(ex);  
 }  
 WSTIMBRADO\_WSDL\_LOCATION = url;  
 WSTIMBRADO\_EXCEPTION = e;  
 }  
  
 public WsTimbrado() {  
 super(\_\_getWsdlLocation(), WSTIMBRADO\_QNAME);  
 }  
  
 public WsTimbrado(WebServiceFeature... features) {  
 super(\_\_getWsdlLocation(), WSTIMBRADO\_QNAME, features);  
 }  
  
 public WsTimbrado(URL wsdlLocation) {  
 super(wsdlLocation, WSTIMBRADO\_QNAME);  
 }  
  
 public WsTimbrado(URL wsdlLocation, WebServiceFeature... features) {  
 super(wsdlLocation, WSTIMBRADO\_QNAME, features);  
 }  
  
 public WsTimbrado(URL wsdlLocation, QName serviceName) {  
 super(wsdlLocation, serviceName);  
 }  
  
 public WsTimbrado(URL wsdlLocation, QName serviceName, WebServiceFeature... features) {  
 super(wsdlLocation, serviceName, features);  
 }  
  
 /\*\*  
 \*  
 \* @return  
 \* returns WsTimbradoSoap  
 \*/  
 @WebEndpoint(name = "wsTimbradoSoap")  
 public WsTimbradoSoap getWsTimbradoSoap() {  
 return super.getPort(new QName("https://cfdi.timbrado.com.mx/timbradov2", "wsTimbradoSoap"),  
 WsTimbradoSoap.class);  
 }  
  
 /\*\*  
 \*  
 \* @param features  
 \* A list of {@link javax.xml.ws.WebServiceFeature} to configure on the proxy. Supported features not in the <code>features</code> parameter will have their default values.  
 \* @return  
 \* returns WsTimbradoSoap  
 \*/  
 @WebEndpoint(name = "wsTimbradoSoap")  
 public WsTimbradoSoap getWsTimbradoSoap(WebServiceFeature... features) {  
 return super.getPort(new QName("https://cfdi.timbrado.com.mx/timbradov2", "wsTimbradoSoap"),  
 WsTimbradoSoap.class, features);  
 }  
  
 /\*\*  
 \*  
 \* @return  
 \* returns WsTimbradoSoap  
 \*/  
 @WebEndpoint(name = "wsTimbradoSoap12")  
 public WsTimbradoSoap getWsTimbradoSoap12() {  
 WsTimbradoSoap soap =  
 super.getPort(new QName("https://cfdi.timbrado.com.mx/timbradov2", "wsTimbradoSoap12"),  
 WsTimbradoSoap.class);  
  
  
 Map<String, Object> req\_ctx = ((BindingProvider) soap).getRequestContext();  
  
  
 AuthenticationHeader ah = new AuthenticationHeader();  
 ah.setUserName("autofac\_t");  
 ah.setPassword("6Pj!N+5sbQ$4=t8Y");  
 //First try  
 /\*  
 List<Header> headersList = new ArrayList<Header>();  
 Map<String, List<String>> headers = new HashMap<String, List<String>>();  
 headers.put("Username", Collections.singletonList("mkyong"));  
 headers.put("Password", Collections.singletonList("password"));  
 req\_ctx.put(MessageContext.HTTP\_REQUEST\_HEADERS, headers);  
  
 Header soapHeader = null;  
 try {  
 soapHeader =  
 new Header(new QName("https://cfdi.timbrado.com.mx/timbradov2", "AuthenticationHeader"), ah,  
 new JAXBDataBinding(AuthenticationHeader.class));  
  
 } catch (JAXBException e) {  
 e.printStackTrace();  
 }  
 headersList.add(soapHeader);  
 req\_ctx.put(Header.HEADER\_LIST, headersList);  
 \*/  
  
 Binding binding = ((BindingProvider) soap).getBinding();  
 final ObjectFactory objectFactory = new ObjectFactory();  
 // creating JAXBElement from headerObj  
 final JAXBElement<AuthenticationHeader> requesterCredentials = objectFactory.createAuthenticationHeader(ah);  
 List<Handler> handlersList = new ArrayList<Handler>();  
 handlersList.add(new SOAPHandler<SOAPMessageContext>() {  
 @Override  
 public boolean handleMessage(final SOAPMessageContext context) {  
 try {  
 // checking whether handled message is outbound one as per Martin Strauss answer  
 final Boolean outbound = (Boolean) context.get("javax.xml.ws.handler.message.outbound");  
 if (outbound != null &&  
 outbound) {  
 // obtaining marshaller which should marshal instance to xml  
 final Marshaller marshaller =  
 JAXBContext.newInstance(AuthenticationHeader.class).createMarshaller();  
 // adding header because otherwise it's null  
 SOAPHeader soapHeader = context.getMessage()  
 .getSOAPPart()  
 .getEnvelope()  
 .getHeader();  
 if (soapHeader == null) {  
 System.out.println("es nulo el header");  
 soapHeader = context.getMessage()  
 .getSOAPPart()  
 .getEnvelope()  
 .addHeader();  
 } else {  
 System.out.println("No es nulo el header: ");  
 System.out.println(soapHeader.toString());  
 }  
  
 // marshalling instance (appending) to SOAP header's xml node  
 marshaller.marshal(requesterCredentials, soapHeader);  
 }  
 } catch (final Exception e) {  
 throw new RuntimeException(e);  
 }  
 return true;  
 }  
  
 // ... default implementations of other methods go here  
  
 @Override  
 public Set<QName> getHeaders() {  
 // TODO Implement this method  
 return Collections.emptySet();  
 }  
  
 @Override  
 public boolean handleFault(SOAPMessageContext sOAPMessageContext) {  
 // TODO Implement this method  
 return false;  
 }  
  
 @Override  
 public void close(MessageContext messageContext) {  
 // TODO Implement this method  
 }  
 });  
  
 // as per Jean-Bernard Pellerin's comment setting handlerChain list here, after all handlers were added to list  
 binding.setHandlerChain(handlersList);  
 return soap;  
  
 }  
  
 /\*\*  
 \*  
 \* @param features  
 \* A list of {@link javax.xml.ws.WebServiceFeature} to configure on the proxy. Supported features not in the <code>features</code> parameter will have their default values.  
 \* @return  
 \* returns WsTimbradoSoap  
 \*/  
 @WebEndpoint(name = "wsTimbradoSoap12")  
 public WsTimbradoSoap getWsTimbradoSoap12(WebServiceFeature... features) {  
 return super.getPort(new QName("https://cfdi.timbrado.com.mx/timbradov2", "wsTimbradoSoap12"),  
 WsTimbradoSoap.class, features);  
 }  
  
 private static URL \_\_getWsdlLocation() {  
 if (WSTIMBRADO\_EXCEPTION != null) {  
 throw WSTIMBRADO\_EXCEPTION;  
 }  
 return WSTIMBRADO\_WSDL\_LOCATION;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\client\facturacion\WsTimbradoSoap.java

package mx.mbi.dgc.client.facturacion;  
  
import javax.jws.WebMethod;  
import javax.jws.WebParam;  
import javax.jws.WebResult;  
import javax.jws.WebService;  
import javax.xml.bind.annotation.XmlSeeAlso;  
import javax.xml.ws.RequestWrapper;  
import javax.xml.ws.ResponseWrapper;  
  
  
/\*\*  
 \* This class was generated by the JAX-WS RI.  
 \* JAX-WS RI 2.2.6-1b01   
 \* Generated source version: 2.2  
 \*   
 \*/  
@WebService(name = "wsTimbradoSoap", targetNamespace = "https://cfdi.timbrado.com.mx/timbradov2")  
@XmlSeeAlso({  
 ObjectFactory.class  
})  
public interface WsTimbradoSoap {  
  
  
 /\*\*  
 \*   
 \* @param cfdiBytes  
 \* @return  
 \* returns mx.mbi.dgc.client.facturacion.Response  
 \*/  
 @WebMethod(operationName = "GeneraTimbre", action = "https://cfdi.timbrado.com.mx/timbradov2/GeneraTimbre")  
 @WebResult(name = "GeneraTimbreResult", targetNamespace = "https://cfdi.timbrado.com.mx/timbradov2")  
 @RequestWrapper(localName = "GeneraTimbre", targetNamespace = "https://cfdi.timbrado.com.mx/timbradov2", className = "mx.mbi.dgc.client.facturacion.GeneraTimbre")  
 @ResponseWrapper(localName = "GeneraTimbreResponse", targetNamespace = "https://cfdi.timbrado.com.mx/timbradov2", className = "mx.mbi.dgc.client.facturacion.GeneraTimbreResponse")  
 public Response generaTimbre(  
 @WebParam(name = "cfdiBytes", targetNamespace = "https://cfdi.timbrado.com.mx/timbradov2")  
 byte[] cfdiBytes);  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatAnioMes.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatAnioMes;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatAnioMes extends JpaRepository<CatAnioMes, java.math.BigDecimal> {  
 Page<CatAnioMes> findByAnioMesId(BigDecimal anioMesId, Pageable page) throws DyCNormaException;  
 Page<CatAnioMes> findByAnioMesContaining(String anioMes, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatCapacidad.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatCapacidad;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatCapacidad extends JpaRepository<CatCapacidad, java.math.BigDecimal> {  
 Page<CatCapacidad> findByCapacidadId(BigDecimal capacidadId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatCargoFijo.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatCargoFijo;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatCargoFijo extends JpaRepository<CatCargoFijo, java.math.BigDecimal> {  
 Page<CatCargoFijo> findByCargoFijoId(BigDecimal cargoFijoId, Pageable page) throws DyCNormaException;  
 Page<CatCargoFijo> findByPathSoporteContaining(String pathSoporte, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatCargos.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatCargos;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatCargos extends JpaRepository<CatCargos, java.math.BigDecimal> {  
 Page<CatCargos> findByCargoId(BigDecimal cargoId, Pageable page) throws DyCNormaException;  
 Page<CatCargos> findByCargoContaining(String cargo, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatClasificacion.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatClasificacion;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatClasificacion extends JpaRepository<CatClasificacion, java.math.BigDecimal> {  
 Page<CatClasificacion> findByClasificacionId(BigDecimal clasificacionId, Pageable page) throws DyCNormaException;  
 Page<CatClasificacion> findByColorContaining(String color, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatClasificacionEvento.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatClasificacionEvento;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOCatClasificacionEvento extends JpaRepository<CatClasificacionEvento, java.math.BigDecimal> {  
 Page<CatClasificacionEvento> findByClasificacionEventoId(BigDecimal clasificacionEventoId,  
 Pageable page) throws DyCNormaException;  
  
 Page<CatClasificacionEvento> findByClasificacionEventoContaining(String clasificacionEvento,  
 Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatClasificacionEventos.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatClasificacionEventos;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatClasificacionEventos extends JpaRepository<CatClasificacionEventos, java.math.BigDecimal> {  
 Page<CatClasificacionEventos> findByClasificacionEventoId(BigDecimal clasificacionEventoId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatClienteMonedas.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatClienteMonedas;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatClienteMonedas extends JpaRepository<CatClienteMonedas, java.math.BigDecimal> {  
 Page<CatClienteMonedas> findByClienteMonedaId(BigDecimal clienteMonedaId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatClientes.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatClientes;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatClientes extends JpaRepository<CatClientes, java.math.BigDecimal> {  
 Page<CatClientes> findByClienteId(BigDecimal clienteId, Pageable page) throws DyCNormaException;  
 Page<CatClientes> findByNombreComercialContaining(String nombreComercial, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatCombustible.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatCombustible;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatCombustible extends JpaRepository<CatCombustible, java.math.BigDecimal> {  
 Page<CatCombustible> findByCombustibleId(BigDecimal combustibleId, Pageable page) throws DyCNormaException;  
 Page<CatCombustible> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatCompania.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatCompania;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatCompania extends JpaRepository<CatCompania, java.math.BigDecimal> {  
 Page<CatCompania> findByCompaniaId(BigDecimal companiaId, Pageable page) throws DyCNormaException;  
 Page<CatCompania> findByNombreComercialContaining(String nombreComercial, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatCompaniaPlantas.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatCompaniaPlantas;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatCompaniaPlantas extends JpaRepository<CatCompaniaPlantas, java.math.BigDecimal> {  
 Page<CatCompaniaPlantas> findByCompaniaPlantaId(BigDecimal companiaPlantaId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatComponentePeriodo.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatComponentePeriodo;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatComponentePeriodo extends JpaRepository<CatComponentePeriodo, java.math.BigDecimal> {  
 Page<CatComponentePeriodo> findByComponentePeriodoId(BigDecimal componentePeriodoId, Pageable page) throws DyCNormaException;  
 Page<CatComponentePeriodo> findByComponentePeriodoContaining(String componentePeriodo, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatComponentes.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatComponentes;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatComponentes extends JpaRepository<CatComponentes, java.math.BigDecimal> {  
 Page<CatComponentes> findByComponenteId(BigDecimal componenteId, Pageable page) throws DyCNormaException;  
 Page<CatComponentes> findByComponenteContaining(String componente, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatCondiciones.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatCondiciones;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatCondiciones extends JpaRepository<CatCondiciones, java.math.BigDecimal> {  
 Page<CatCondiciones> findByCondicionId(BigDecimal condicionId, Pageable page) throws DyCNormaException;  
 Page<CatCondiciones> findByKeyCondicionContaining(String keyCondicion, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatConexiones.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatConexiones;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatConexiones extends JpaRepository<CatConexiones, java.math.BigDecimal> {  
 Page<CatConexiones> findByConexionId(BigDecimal conexionId, Pageable page) throws DyCNormaException;  
 Page<CatConexiones> findByConexionContaining(String conexion, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatConfiguracionDetalle.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatConfiguracionDetalle;  
import mx.mbi.dgc.jpa.entity.CatConfiguracionEventos;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOCatConfiguracionDetalle extends JpaRepository<CatConfiguracionDetalle, java.math.BigDecimal> {  
 Page<CatConfiguracionDetalle> findByConfiguracionDetalleId(BigDecimal configuracionDetalleId,  
 Pageable page) throws DyCNormaException;  
  
  
 List<CatConfiguracionDetalle> findByConfiguracionEventoId(CatConfiguracionEventos configuracionEventoId) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatConfiguracionEventos.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatClasificacionEvento;  
import mx.mbi.dgc.jpa.entity.CatConfiguracionEventos;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOCatConfiguracionEventos extends JpaRepository<CatConfiguracionEventos, java.math.BigDecimal> {  
 Page<CatConfiguracionEventos> findByConfiguracionEventoId(BigDecimal configuracionEventoId,  
 Pageable page) throws DyCNormaException;  
  
 Page<CatConfiguracionEventos> findByConfiguracionContaining(String configuracion,  
 Pageable page) throws DyCNormaException;  
  
 List<CatConfiguracionEventos> findByClasificacionEventoId(CatClasificacionEvento clasificacionEventoId) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatConfiguracionOpciones.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatConfiguracionDetalle;  
import mx.mbi.dgc.jpa.entity.CatConfiguracionOpciones;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOCatConfiguracionOpciones extends JpaRepository<CatConfiguracionOpciones, java.math.BigDecimal> {  
 Page<CatConfiguracionOpciones> findByConfiguracionOpcionesId(BigDecimal configuracionOpcionesId,  
 Pageable page) throws DyCNormaException;  
  
  
 List<CatConfiguracionOpciones> findByConfiguracionDetalleId(CatConfiguracionDetalle configuracionDetalleId) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatContratosAfectados.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatContratosAfectados;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatContratosAfectados extends JpaRepository<CatContratosAfectados, java.math.BigDecimal> {  
 Page<CatContratosAfectados> findByContratoAfectadoId(BigDecimal contratoAfectadoId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatCurvasCorreccion.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatCurvasCorreccion;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatCurvasCorreccion extends JpaRepository<CatCurvasCorreccion, java.math.BigDecimal> {  
 Page<CatCurvasCorreccion> findByCurvaId(BigDecimal curvaId, Pageable page) throws DyCNormaException;  
 Page<CatCurvasCorreccion> findByFxContaining(String fx, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatDatosCertificados.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatDatosCertificados;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatDatosCertificados extends JpaRepository<CatDatosCertificados, java.math.BigDecimal> {  
 Page<CatDatosCertificados> findByDatosCertificadoId(BigDecimal datosCertificadoId, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatDatosFiscales.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatDatosFiscales;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatDatosFiscales extends JpaRepository<CatDatosFiscales, java.math.BigDecimal> {  
 Page<CatDatosFiscales> findByDatosFiscalesId(BigDecimal datosFiscalesId, Pageable page) throws DyCNormaException;  
 Page<CatDatosFiscales> findByCalleContaining(String calle, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatDetalleOpciones.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatDetalleOpciones;  
import mx.mbi.dgc.jpa.entity.CatDetalles;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
  
@Repository  
public interface DAOCatDetalleOpciones extends JpaRepository<CatDetalleOpciones, java.math.BigDecimal> {  
 List<CatDetalleOpciones> findByDetalleId(CatDetalles catDetalles) throws DyCNormaException;  
 Page<CatDetalleOpciones> findByDetalleOpcionesId(BigDecimal detalleOpcionesId, Pageable page) throws DyCNormaException;  
 Page<CatDetalleOpciones> findByOpcionContaining(String opcion, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatDetalles.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatDetalles;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
  
@Repository  
public interface DAOCatDetalles extends JpaRepository<CatDetalles, java.math.BigDecimal> {  
 CatDetalles findByDetalle(String detalle) throws DyCNormaException;  
 CatDetalles findByDetalleId(BigDecimal detalleId) throws DyCNormaException;  
 Page<CatDetalles> findByDetalleId(BigDecimal detalleId, Pageable page) throws DyCNormaException;  
 Page<CatDetalles> findByDetalleContaining(String detalle, Pageable page) throws DyCNormaException;  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatEstados.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatEstados;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatEstados extends JpaRepository<CatEstados, java.math.BigDecimal> {  
 Page<CatEstados> findByEstadoId(BigDecimal estadoId, Pageable page) throws DyCNormaException;  
 Page<CatEstados> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatEstatusBitacora.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatEstatusBitacora;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatEstatusBitacora extends JpaRepository<CatEstatusBitacora, java.math.BigDecimal> {  
 Page<CatEstatusBitacora> findByEstatusBitacoraId(BigDecimal estatusBitacoraId, Pageable page) throws DyCNormaException;  
 Page<CatEstatusBitacora> findByEstatusContaining(String estatus, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatEstatusFactura.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatEstatusFactura;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatEstatusFactura extends JpaRepository<CatEstatusFactura, java.math.BigDecimal> {  
 Page<CatEstatusFactura> findByEstatusFacturaId(BigDecimal estatusFacturaId, Pageable page) throws DyCNormaException;  
 Page<CatEstatusFactura> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatEstatusPagos.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatEstatusPagos;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatEstatusPagos extends JpaRepository<CatEstatusPagos, java.math.BigDecimal> {  
 Page<CatEstatusPagos> findByEstatusPagoId(BigDecimal estatusPagoId, Pageable page) throws DyCNormaException;  
 Page<CatEstatusPagos> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatEstatusPerfil.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatEstatusPerfil;  
  
import mx.mbi.dgc.jpa.entity.CatEstatusVariable;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
import java.util.List;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatEstatusPerfil extends JpaRepository<CatEstatusPerfil, java.math.BigDecimal> {  
 Page<CatEstatusPerfil> findByEstatusPerfilId(BigDecimal estatusPerfilId, Pageable page) throws DyCNormaException;  
 Page<CatEstatusPerfil> findByEstatusPerfilContaining(String estatusPerfil, Pageable page) throws DyCNormaException;  
  
 @Query("SELECT c FROM CatEstatusPerfil c")  
 List<CatEstatusPerfil> getCatEstatusPerfilByEstatusPerfilId();  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatEstatusVariable.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatEstatusVariable;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
import java.util.List;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatEstatusVariable extends JpaRepository<CatEstatusVariable, java.math.BigDecimal> {  
 Page<CatEstatusVariable> findByEstatusVarId(BigDecimal estatusVarId, Pageable page) throws DyCNormaException;  
 Page<CatEstatusVariable> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
  
 @Query("SELECT c FROM CatEstatusVariable c")  
 List<CatEstatusVariable> getCatEstatusVariableByEstatusVariableId();  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatEtapas.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatEtapas;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatEtapas extends JpaRepository<CatEtapas, java.math.BigDecimal> {  
 Page<CatEtapas> findByEtapaId(BigDecimal etapaId, Pageable page) throws DyCNormaException;  
 Page<CatEtapas> findByEtapaContaining(String etapa, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatEventos.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatEventos;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatEventos extends JpaRepository<CatEventos, java.math.BigDecimal> {  
 Page<CatEventos> findByEventoId(BigDecimal eventoId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatFactorTipo.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatFactorTipo;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatFactorTipo extends JpaRepository<CatFactorTipo, java.math.BigDecimal> {  
 Page<CatFactorTipo> findByFactorTipoId(BigDecimal factorTipoId, Pageable page) throws DyCNormaException;  
 Page<CatFactorTipo> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatFacturaProductos.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatFacturaProductos;  
import mx.mbi.dgc.jpa.entity.TxDocumentos;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOCatFacturaProductos extends JpaRepository<CatFacturaProductos, java.math.BigDecimal> {  
 Page<CatFacturaProductos> findByFacturaProductoId(BigDecimal facturaProductoId,  
 Pageable page) throws DyCNormaException;  
  
 Page<CatFacturaProductos> findByObservacionesContaining(String observaciones,  
 Pageable page) throws DyCNormaException;  
  
 List<CatFacturaProductos> findByTxDocumentos(TxDocumentos txDocumentos);  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatFirmaCincominutal.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatFirmaCincominutal;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatFirmaCincominutal extends JpaRepository<CatFirmaCincominutal, java.math.BigDecimal> {  
 Page<CatFirmaCincominutal> findByFirmaId(BigDecimal firmaId, Pageable page) throws DyCNormaException;  
 Page<CatFirmaCincominutal> findByFirmaContaining(String firma, Pageable page) throws DyCNormaException;  
 List<CatFirmaCincominutal> findFirst288ByOrderByFirmaId()throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatFormasPago.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatFormasPago;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatFormasPago extends JpaRepository<CatFormasPago, java.math.BigDecimal> {  
 Page<CatFormasPago> findByFormaPagoId(BigDecimal formaPagoId, Pageable page) throws DyCNormaException;  
 Page<CatFormasPago> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatFormulaComponentes.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatFormulaComponentes;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatFormulaComponentes extends JpaRepository<CatFormulaComponentes, java.math.BigDecimal> {  
 Page<CatFormulaComponentes> findByCatFormulaComponentesId(BigDecimal catFormulaComponentesId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatFormulaComponentesTxEvento.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatFormulaComponentesTxEvento;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatFormulaComponentesTxEvento extends JpaRepository<CatFormulaComponentesTxEvento, java.math.BigDecimal> {  
 Page<CatFormulaComponentesTxEvento> findByCatFormulaComponentesTxEventoId(BigDecimal catFormulaComponentesTxEventoId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatImpuestos.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatImpuestos;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatImpuestos extends JpaRepository<CatImpuestos, java.math.BigDecimal> {  
 Page<CatImpuestos> findByImpuestoId(BigDecimal impuestoId, Pageable page) throws DyCNormaException;  
 Page<CatImpuestos> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatLimitesDominio.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatLimitesDominio;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatLimitesDominio extends JpaRepository<CatLimitesDominio, java.math.BigDecimal> {  
 Page<CatLimitesDominio> findByLimitesDominioId(BigDecimal limitesDominioId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatMedidor.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatMedidor;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatMedidor extends JpaRepository<CatMedidor, java.math.BigDecimal> {  
 Page<CatMedidor> findByMedidorId(BigDecimal medidorId, Pageable page) throws DyCNormaException;  
 Page<CatMedidor> findByMedidorContaining(String medidor, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatMedidores.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatMedidores;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatMedidores extends JpaRepository<CatMedidores, java.math.BigDecimal> {  
 Page<CatMedidores> findByMedidorId(BigDecimal medidorId, Pageable page) throws DyCNormaException;  
 Page<CatMedidores> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatMesCargo.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatMesCargo;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatMesCargo extends JpaRepository<CatMesCargo, java.math.BigDecimal> {  
 Page<CatMesCargo> findByMesId(BigDecimal mesId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatMesDia.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatMesDia;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatMesDia extends JpaRepository<CatMesDia, java.math.BigDecimal> {  
 Page<CatMesDia> findByMesDiaId(BigDecimal mesDiaId, Pageable page) throws DyCNormaException;  
 Page<CatMesDia> findByDescripcionContaining(String descripcion, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatMesDiaFirma.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatMesDiaFirma;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatMesDiaFirma extends JpaRepository<CatMesDiaFirma, java.math.BigDecimal> {  
 Page<CatMesDiaFirma> findByMesDiaFirmaId(BigDecimal mesDiaFirmaId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatMetodoPago.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatMetodoPago;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatMetodoPago extends JpaRepository<CatMetodoPago, java.math.BigDecimal> {  
 Page<CatMetodoPago> findByMetodoPagoId(BigDecimal metodoPagoId, Pageable page) throws DyCNormaException;  
 Page<CatMetodoPago> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatMonedas.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatMonedas;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatMonedas extends JpaRepository<CatMonedas, java.math.BigDecimal> {  
 Page<CatMonedas> findByMonedaId(BigDecimal monedaId, Pageable page) throws DyCNormaException;  
 Page<CatMonedas> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatOpcionPlanta.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatCapacidad;  
import mx.mbi.dgc.jpa.entity.CatDetalleOpciones;  
import mx.mbi.dgc.jpa.entity.CatOpcionPlanta;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOCatOpcionPlanta extends JpaRepository<CatOpcionPlanta, java.math.BigDecimal> {  
 List<CatOpcionPlanta> findByPlantaId(CatPlanta plantaId) throws DyCNormaException;  
 List<CatOpcionPlanta> findByDetalleOpcionesId (CatDetalleOpciones detalleOpcionesId) throws DyCNormaException;  
 Page<CatOpcionPlanta> findByOpcionPlantaId(BigDecimal opcionPlantaId, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatPaises.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatPaises;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatPaises extends JpaRepository<CatPaises, java.math.BigDecimal> {  
 Page<CatPaises> findByPaisId(BigDecimal paisId, Pageable page) throws DyCNormaException;  
 Page<CatPaises> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatPermiso.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatPermiso;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatPermiso extends JpaRepository<CatPermiso, java.math.BigDecimal> {  
 Page<CatPermiso> findByPermisoId(BigDecimal permisoId, Pageable page) throws DyCNormaException;  
 Page<CatPermiso> findByPermisoContaining(String permiso, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatPlanta.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatPlanta extends JpaRepository<CatPlanta, java.math.BigDecimal> {  
 Page<CatPlanta> findByPlantaId(BigDecimal plantaId, Pageable page) throws DyCNormaException;  
 Page<CatPlanta> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatPlantaTipoFacturas.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.jpa.entity.CatPlantaTipoFacturas;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOCatPlantaTipoFacturas extends JpaRepository<CatPlantaTipoFacturas, java.math.BigDecimal> {  
 Page<CatPlantaTipoFacturas> findByPlantaTipoFacturaId(BigDecimal plantaTipoFacturaId,  
 Pageable page) throws DyCNormaException;  
  
 List<CatPlantaTipoFacturas> findByCatPlanta(CatPlanta catPlanta);  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatPropCincominutal.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatPropCincominutal;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatPropCincominutal extends JpaRepository<CatPropCincominutal, java.math.BigDecimal> {  
 Page<CatPropCincominutal> findByPropiedadId(BigDecimal propiedadId, Pageable page) throws DyCNormaException;  
 Page<CatPropCincominutal> findByPropiedadContaining(String propiedad, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatQuery.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatQuery;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatQuery extends JpaRepository<CatQuery, java.math.BigDecimal> {  
 Page<CatQuery> findByQueryId(BigDecimal queryId, Pageable page) throws DyCNormaException;  
 Page<CatQuery> findByQueryContaining(String query, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatRangoVariable.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatRangoVariable;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatRangoVariable extends JpaRepository<CatRangoVariable, java.math.BigDecimal> {  
 Page<CatRangoVariable> findByRangoId(BigDecimal rangoId, Pageable page) throws DyCNormaException;  
 Page<CatRangoVariable> findByUnificacionEscalaContaining(String unificacionEscala, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatRegimenFiscal.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatRegimenFiscal;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatRegimenFiscal extends JpaRepository<CatRegimenFiscal, java.math.BigDecimal> {  
 Page<CatRegimenFiscal> findByRegimenFiscalId(BigDecimal regimenFiscalId, Pageable page) throws DyCNormaException;  
 Page<CatRegimenFiscal> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatReglaNegocio.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatReglaNegocio;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatReglaNegocio extends JpaRepository<CatReglaNegocio, java.math.BigDecimal> {  
 Page<CatReglaNegocio> findByReglaNegocioId(BigDecimal reglaNegocioId, Pageable page) throws DyCNormaException;  
 Page<CatReglaNegocio> findByAlgoritmoContaining(String algoritmo, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatRol.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatRol;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatRol extends JpaRepository<CatRol, java.math.BigDecimal> {  
 Page<CatRol> findByRolId(BigDecimal rolId, Pageable page) throws DyCNormaException;  
 Page<CatRol> findByRolContaining(String rol, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatServicioImpuestos.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatServicioImpuestos;  
import mx.mbi.dgc.jpa.entity.CatServicios;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOCatServicioImpuestos extends JpaRepository<CatServicioImpuestos, java.math.BigDecimal> {  
 Page<CatServicioImpuestos> findByServicioImpuestoId(BigDecimal servicioImpuestoId,  
 Pageable page) throws DyCNormaException;  
  
 List<CatServicioImpuestos> findByCatServicios(CatServicios catServicios);  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatServicios.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatServicios;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatServicios extends JpaRepository<CatServicios, java.math.BigDecimal> {  
 Page<CatServicios> findByServicioId(BigDecimal servicioId, Pageable page) throws DyCNormaException;  
 Page<CatServicios> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatSistemas.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatSistemas ;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatSistemas extends JpaRepository<CatSistemas , java.math.BigDecimal> {  
 Page<CatSistemas > findBySistemaId(BigDecimal sistemaId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatSubCargos.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatSubCargos;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatSubCargos extends JpaRepository<CatSubCargos, java.math.BigDecimal> {  
 Page<CatSubCargos> findBySubCargoId(BigDecimal subCargoId, Pageable page) throws DyCNormaException;  
 Page<CatSubCargos> findBySubCargoContaining(String subCargo, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatTags.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
import java.math.BigInteger;  
  
import java.util.Date;  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatTags;  
  
import mx.mbi.dgc.jpa.entity.TxCincominutalVar;  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.data.repository.query.Param;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOCatTags extends JpaRepository<CatTags, BigDecimal> {  
  
 @Query("Select ct from CatTags ct where ct.plantaId = :plantaId ")  
 List<CatTags> getCatTagsByPlantaId(@Param("plantaId") BigDecimal plantaId);  
  
 Page<CatTags> findByTagId(BigDecimal tagId, Pageable page) throws DyCNormaException;  
  
 Page<CatTags> findByTagContaining(String tag, Pageable page) throws DyCNormaException;  
  
 List<CatTags> findByPadreIdAndPlantaId(BigDecimal padreId, BigDecimal plantaId) throws DyCNormaException;  
  
 List<CatTags> findByPlantaIdOrderByOrdenAsc(BigDecimal plantaId) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatTipoDato.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatTipoDato;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOCatTipoDato extends JpaRepository<CatTipoDato, java.math.BigDecimal> {  
 Page<CatTipoDato> findByTipoDatoId(BigDecimal tipoDatoId, Pageable page) throws DyCNormaException;  
 Page<CatTipoDato> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatTipoDocumento.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatTipoDocumento;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatTipoDocumento extends JpaRepository<CatTipoDocumento, java.math.BigDecimal> {  
 Page<CatTipoDocumento> findByTipoDocumentoId(BigDecimal tipoDocumentoId, Pageable page) throws DyCNormaException;  
 Page<CatTipoDocumento> findByTipoContaining(String tipo, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatTipoFactoresSat.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatTipoFactoresSat;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatTipoFactoresSat extends JpaRepository<CatTipoFactoresSat, java.math.BigDecimal> {  
 Page<CatTipoFactoresSat> findByTipoFactorSatId(BigDecimal tipoFactorSatId, Pageable page) throws DyCNormaException;  
 Page<CatTipoFactoresSat> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatTipoImpuestos.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatTipoImpuestos;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatTipoImpuestos extends JpaRepository<CatTipoImpuestos, java.math.BigDecimal> {  
 Page<CatTipoImpuestos> findByTipoImpuestoId(BigDecimal tipoImpuestoId, Pageable page) throws DyCNormaException;  
 Page<CatTipoImpuestos> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatTipoRelacion.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatTipoRelacion;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatTipoRelacion extends JpaRepository<CatTipoRelacion, java.math.BigDecimal> {  
 Page<CatTipoRelacion> findByTipoRelacionId(BigDecimal tipoRelacionId, Pageable page) throws DyCNormaException;  
 Page<CatTipoRelacion> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatTiposDeEvento.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatTiposDeEvento;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatTiposDeEvento extends JpaRepository<CatTiposDeEvento, java.math.BigDecimal> {  
 Page<CatTiposDeEvento> findByTiposDeEventoId(BigDecimal tiposDeEventoId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatTiposFacturas.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatTiposFacturas;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatTiposFacturas extends JpaRepository<CatTiposFacturas, java.math.BigDecimal> {  
 Page<CatTiposFacturas> findByTipoFacturaId(BigDecimal tipoFacturaId, Pageable page) throws DyCNormaException;  
 Page<CatTiposFacturas> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatTiposServicios.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatTiposServicios;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatTiposServicios extends JpaRepository<CatTiposServicios, java.math.BigDecimal> {  
 Page<CatTiposServicios> findByTipoServicioId(BigDecimal tipoServicioId, Pageable page) throws DyCNormaException;  
 Page<CatTiposServicios> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatUnidad.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatUnidad;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatUnidad extends JpaRepository<CatUnidad, java.math.BigDecimal> {  
 Page<CatUnidad> findByUnidadId(BigDecimal unidadId, Pageable page) throws DyCNormaException;  
 Page<CatUnidad> findByUnidadContaining(String unidad, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatUnidades.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatUnidades ;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatUnidades extends JpaRepository<CatUnidades , java.math.BigDecimal> {  
 Page<CatUnidades > findByUnidadId(BigDecimal unidadId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatUsoCfdi.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatUsoCfdi;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatUsoCfdi extends JpaRepository<CatUsoCfdi, java.math.BigDecimal> {  
 Page<CatUsoCfdi> findByUsoCfdiId(BigDecimal usoCfdiId, Pageable page) throws DyCNormaException;  
 Page<CatUsoCfdi> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatVariable.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatVariable;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatVariable extends JpaRepository<CatVariable, java.math.BigDecimal> {  
 Page<CatVariable> findByVariableId(BigDecimal variableId, Pageable page) throws DyCNormaException;  
 Page<CatVariable> findByUnidadMedidaContaining(String unidadMedida, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatVariablePerfil.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatVariablePerfil;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatVariablePerfil extends JpaRepository<CatVariablePerfil, java.math.BigDecimal> {  
 Page<CatVariablePerfil> findByVariablePerfilId(BigDecimal variablePerfilId, Pageable page) throws DyCNormaException;  
 Page<CatVariablePerfil> findByVariablePerfilContaining(String variablePerfil, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOCatVersion.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatVersion;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOCatVersion extends JpaRepository<CatVersion, java.math.BigDecimal> {  
 Page<CatVersion> findByVersionId(BigDecimal versionId, Pageable page) throws DyCNormaException;  
 Page<CatVersion> findByVersionContaining(String version, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOSatImpuestos.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.SatImpuestos;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOSatImpuestos extends JpaRepository<SatImpuestos, java.math.BigDecimal> {  
 Page<SatImpuestos> findByImpuestoId(BigDecimal impuestoId, Pageable page) throws DyCNormaException;  
 Page<SatImpuestos> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOSatMonedas.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.SatMonedas;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOSatMonedas extends JpaRepository<SatMonedas, java.math.BigDecimal> {  
 Page<SatMonedas> findByMonedaSatId(BigDecimal monedaSatId, Pageable page) throws DyCNormaException;  
 Page<SatMonedas> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOSatPaises.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.SatPaises;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOSatPaises extends JpaRepository<SatPaises, java.math.BigDecimal> {  
 Page<SatPaises> findByPaisSatId(BigDecimal paisSatId, Pageable page) throws DyCNormaException;  
 Page<SatPaises> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOSatProductosClaves.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.SatProductosClaves;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOSatProductosClaves extends JpaRepository<SatProductosClaves, java.math.BigDecimal> {  
 Page<SatProductosClaves> findBySatProductosClavesId(BigDecimal satProductosClavesId, Pageable page) throws DyCNormaException;  
 Page<SatProductosClaves> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOSatProductosUnidades.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.SatProductosUnidades;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOSatProductosUnidades extends JpaRepository<SatProductosUnidades, java.math.BigDecimal> {  
 Page<SatProductosUnidades> findBySatProductosUnidadesId(BigDecimal satProductosUnidadesId, Pageable page) throws DyCNormaException;  
 Page<SatProductosUnidades> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOSatTasaCuota.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.SatTasaCuota;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOSatTasaCuota extends JpaRepository<SatTasaCuota, java.math.BigDecimal> {  
 Page<SatTasaCuota> findByTasaCuotaId(BigDecimal tasaCuotaId, Pageable page) throws DyCNormaException;  
 Page<SatTasaCuota> findByRangoContaining(String rango, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOSatTipoFactor.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.SatTipoFactor;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOSatTipoFactor extends JpaRepository<SatTipoFactor, java.math.BigDecimal> {  
 Page<SatTipoFactor> findByTipoFactorId(BigDecimal tipoFactorId, Pageable page) throws DyCNormaException;  
 Page<SatTipoFactor> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOSatTipoRelacion.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.SatTipoRelacion;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOSatTipoRelacion extends JpaRepository<SatTipoRelacion, java.lang.String> {  
 Page<SatTipoRelacion> findByNombre(BigDecimal nombre, Pageable page) throws DyCNormaException;  
 Page<SatTipoRelacion> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOSatUsoCfdi.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.SatUsoCfdi;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOSatUsoCfdi extends JpaRepository<SatUsoCfdi, java.lang.String> {  
 Page<SatUsoCfdi> findByDescripcion(BigDecimal descripcion, Pageable page) throws DyCNormaException;  
 Page<SatUsoCfdi> findByDescripcionContaining(String descripcion, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxAdjuntos.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxAdjuntos;  
import mx.mbi.dgc.jpa.entity.TxEvento;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOTxAdjuntos extends JpaRepository<TxAdjuntos, java.math.BigDecimal> {  
 Page<TxAdjuntos> findByAdjuntoId(BigDecimal adjuntoId, Pageable page) throws DyCNormaException;  
  
 Page<TxAdjuntos> findByReferenciaContaining(String referencia, Pageable page) throws DyCNormaException;  
  
 List<TxAdjuntos> findByEventoId(TxEvento eventoId) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxBitacora.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import java.util.Date;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxBitacora;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOTxBitacora extends JpaRepository<TxBitacora, java.math.BigDecimal> {  
 Page<TxBitacora> findByBitacoraId(BigDecimal bitacoraId, Pageable page) throws DyCNormaException;  
  
 TxBitacora findByFecha(Date fecha) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxCincominutal.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxCincominutal;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.data.repository.query.Param;  
  
@Repository  
public interface DAOTxCincominutal extends JpaRepository<TxCincominutal, java.math.BigDecimal> {  
 Page<TxCincominutal> findByCincominutalId(BigDecimal cincominutalId, Pageable page) throws DyCNormaException;  
   
 @Query("select txCincominutal FROM TxCincominutal txCincominutal " +  
 "RIGHT JOIN FETCH txCincominutal.txCincominutalVars cincoVar " +  
 "LEFT JOIN FETCH cincoVar.catVariable catv " +  
 "LEFT JOIN FETCH cincoVar.txCincominutal txc " +   
 "LEFT JOIN FETCH txCincominutal.catMesDiaFirma cmdf " +  
 "LEFT JOIN FETCH cmdf.catFirmaCincominutal " +   
 "LEFT JOIN FETCH cmdf.catMesDia ctmd " +  
 "LEFT JOIN FETCH ctmd.catAnioMes cam " +   
 "where " +  
 "txCincominutal.cincominutalId = :cincominutalId")  
 public TxCincominutal obtenTxCincominutal(@Param("cincominutalId")BigDecimal cincominutalId);  
   
   
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxCincominutalVar.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.jpa.entity.TxCincominutal;  
import mx.mbi.dgc.jpa.entity.TxCincominutalVar;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOTxCincominutalVar extends JpaRepository<TxCincominutalVar, java.math.BigDecimal> {  
 Page<TxCincominutalVar> findByCincominutalVarId(BigDecimal cincominutalVarId,  
 Pageable page) throws DyCNormaException;  
  
 Page<TxCincominutalVar> findByValorOriginalContaining(String valorOriginal, Pageable page) throws DyCNormaException;  
  
 List<TxCincominutalVar> findByEstatusCnNotAndFcPrimeraIsNullAndTxCincominutal\_CatPlanta(BigDecimal estatusCn,CatPlanta catPlanta) throws DyCNormaException;  
  
 List<TxCincominutalVar> findByTxCincominutal(TxCincominutal txCincominutal) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxCrudos.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxCrudos;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxCrudos extends JpaRepository<TxCrudos, java.math.BigDecimal> {  
 Page<TxCrudos> findByDia(BigDecimal dia, Pageable page) throws DyCNormaException;  
 Page<TxCrudos> findByDiaContaining(String dia, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxCrudosLimpios.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxCrudosLimpios;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxCrudosLimpios extends JpaRepository<TxCrudosLimpios, java.lang.String> {  
 Page<TxCrudosLimpios> findByDia(BigDecimal dia, Pageable page) throws DyCNormaException;  
 Page<TxCrudosLimpios> findByMinutoContaining(String minuto, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxCrudosSol.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxCrudosSol;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxCrudosSol extends JpaRepository<TxCrudosSol, java.math.BigDecimal> {  
 Page<TxCrudosSol> findByDia(BigDecimal dia, Pageable page) throws DyCNormaException;  
 Page<TxCrudosSol> findByDiaContaining(String dia, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxDocumentos.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxDocumentos;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxDocumentos extends JpaRepository<TxDocumentos, java.math.BigDecimal> {  
 Page<TxDocumentos> findByDocumentoId(BigDecimal documentoId, Pageable page) throws DyCNormaException;  
 Page<TxDocumentos> findBySerieContaining(String serie, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxErrorCargaCrudos.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxErrorCargaCrudos;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxErrorCargaCrudos extends JpaRepository<TxErrorCargaCrudos, java.math.BigDecimal> {  
 Page<TxErrorCargaCrudos> findByTipoError(BigDecimal tipoError, Pageable page) throws DyCNormaException;  
 Page<TxErrorCargaCrudos> findByErrmContaining(String errm, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxErrorFormato.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxErrorFormato;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxErrorFormato extends JpaRepository<TxErrorFormato, java.math.BigDecimal> {  
 Page<TxErrorFormato> findByErrorFormatoId(BigDecimal errorFormatoId, Pageable page) throws DyCNormaException;  
 Page<TxErrorFormato> findByValorContaining(String valor, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxEscenario.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxEscenario;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxEscenario extends JpaRepository<TxEscenario, java.math.BigDecimal> {  
 Page<TxEscenario> findByEscenarioId(BigDecimal escenarioId, Pageable page) throws DyCNormaException;  
 Page<TxEscenario> findByNombreContaining(String nombre, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxEscenariosBitacora.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxEscenariosBitacora;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxEscenariosBitacora extends JpaRepository<TxEscenariosBitacora, java.math.BigDecimal> {  
 Page<TxEscenariosBitacora> findByEscenarioBitacoraId(BigDecimal escenarioBitacoraId, Pageable page) throws DyCNormaException;  
 Page<TxEscenariosBitacora> findByEscenarioContaining(String escenario, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxEscenarioVariable.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxEscenarioVariable;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxEscenarioVariable extends JpaRepository<TxEscenarioVariable, java.math.BigDecimal> {  
 Page<TxEscenarioVariable> findByEscenarioVariableId(BigDecimal escenarioVariableId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxEvento.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxBitacora;  
import mx.mbi.dgc.jpa.entity.TxEvento;  
import mx.mbi.dgc.jpa.entity.TxUsuario;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOTxEvento extends JpaRepository<TxEvento, java.math.BigDecimal> {  
 Page<TxEvento> findByEventoId(BigDecimal eventoId, Pageable page) throws DyCNormaException;  
  
 Page<TxEvento> findByInformesContaining(String informes, Pageable page) throws DyCNormaException;  
  
 List<TxEvento> findByBitacoraIdAndUsuarioId(TxBitacora bitacoraId, TxUsuario usuarioId) throws DyCNormaException;  
  
 List<TxEvento> findByClasificacionEventoId\_ClasificacionEventoId(BigDecimal clasificacionEventoId);  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxEventoDetalles.java

package mx.mbi.dgc.dao;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxEvento;  
import mx.mbi.dgc.jpa.entity.TxEventoDetalles;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOTxEventoDetalles extends JpaRepository<TxEventoDetalles, java.math.BigDecimal> {  
 Page<TxEventoDetalles> findByEventoDetalleId(BigDecimal eventoDetalleId, Pageable page) throws DyCNormaException;  
 List<TxEventoDetalles> findByEventoId(TxEvento eventoId)throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxFacturaUuids.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxFacturaUuids;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxFacturaUuids extends JpaRepository<TxFacturaUuids, java.math.BigDecimal> {  
 Page<TxFacturaUuids> findByFacturaUuidId(BigDecimal facturaUuidId, Pageable page) throws DyCNormaException;  
 Page<TxFacturaUuids> findByObservacionesContaining(String observaciones, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxFormulas.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxFormulas;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxFormulas extends JpaRepository<TxFormulas, java.math.BigDecimal> {  
 Page<TxFormulas> findByFormulaId(BigDecimal formulaId, Pageable page) throws DyCNormaException;  
 Page<TxFormulas> findByFormulaContaining(String formula, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxLogErrorCarga.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxLogErrorCarga;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxLogErrorCarga extends JpaRepository<TxLogErrorCarga, java.math.BigDecimal> {  
 Page<TxLogErrorCarga> findByTableName(BigDecimal tableName, Pageable page) throws DyCNormaException;  
 Page<TxLogErrorCarga> findByTableNameContaining(String tableName, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxPerfilesVariable.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxPerfilesVariable;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxPerfilesVariable extends JpaRepository<TxPerfilesVariable, java.math.BigDecimal> {  
 Page<TxPerfilesVariable> findByPerfilId(BigDecimal perfilId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxProduccion.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxProduccion;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxProduccion extends JpaRepository<TxProduccion, java.math.BigDecimal> {  
 Page<TxProduccion> findByProduccionId(BigDecimal produccionId, Pageable page) throws DyCNormaException;  
 Page<TxProduccion> findByValorContaining(String valor, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxRolPermiso.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxRolPermiso;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxRolPermiso extends JpaRepository<TxRolPermiso, java.math.BigDecimal> {  
 Page<TxRolPermiso> findByRolPermisoId(BigDecimal rolPermisoId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxTrazabilidad.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxTrazabilidad;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxTrazabilidad extends JpaRepository<TxTrazabilidad, java.math.BigDecimal> {  
 Page<TxTrazabilidad> findByTrazabilidadId(BigDecimal trazabilidadId, Pageable page) throws DyCNormaException;  
 Page<TxTrazabilidad> findByAccionContaining(String accion, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxTrazaQuery.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxTrazaQuery;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxTrazaQuery extends JpaRepository<TxTrazaQuery, java.math.BigDecimal> {  
 Page<TxTrazaQuery> findByTrazaQueryId(BigDecimal trazaQueryId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxUsuario.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxUsuario;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxUsuario extends JpaRepository<TxUsuario, java.math.BigDecimal> {  
 Page<TxUsuario> findByUsuarioId(BigDecimal usuarioId, Pageable page) throws DyCNormaException;  
 Page<TxUsuario> findByUsernameContaining(String username, Pageable page) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxUsuarioPlanta.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxUsuarioPlanta;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxUsuarioPlanta extends JpaRepository<TxUsuarioPlanta, java.math.BigDecimal> {  
 Page<TxUsuarioPlanta> findByUsuarioPlantaId(BigDecimal usuarioPlantaId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxUsuarioRol.java

package mx.mbi.dgc.dao;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxUsuarioRol;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface DAOTxUsuarioRol extends JpaRepository<TxUsuarioRol, java.math.BigDecimal> {  
 Page<TxUsuarioRol> findByUsuarioRolId(BigDecimal usuarioRolId, Pageable page) throws DyCNormaException;  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\DAOTxValorFacturacion.java

package mx.mbi.dgc.dao;  
  
import java.util.Date;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxValorFActuracion;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOTxValorFacturacion extends JpaRepository<TxValorFActuracion, java.math.BigDecimal> {  
 TxValorFActuracion findByVariableFacturacion\_Nombre(String nombre) throws DyCNormaException;  
 TxValorFActuracion findByVariableFacturacion\_NombreAndFecha(String nombre,Date fecha) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\cargainicial\CargaInformacionImpl.java

package mx.mbi.dgc.dao.negocio.cargainicial;  
  
import javax.persistence.EntityManager;  
import javax.persistence.ParameterMode;  
import javax.persistence.PersistenceContext;  
import javax.persistence.StoredProcedureQuery;  
  
import org.springframework.scheduling.annotation.Async;  
import org.springframework.stereotype.Component;  
import org.springframework.transaction.annotation.Transactional;  
  
@Component  
public class CargaInformacionImpl implements ICargaInformacion {  
 @PersistenceContext  
 private EntityManager em;  
  
 @Async("threadPoolTaskExecutor")  
 @Transactional(timeout = 600)  
 @Override  
 public boolean procesaFile(String procedureName, String table, String dir, String filename, Integer ignore,  
 Integer typeload, Integer idUser) {  
 System.out.println("invloke procesaFile");  
 boolean retorno = true;  
 try {  
 StoredProcedureQuery storedProcedureQuery = em.createStoredProcedureQuery(procedureName);  
  
 // Registrar los parámetros de entrada y salida  
 storedProcedureQuery.registerStoredProcedureParameter("P\_TABLE", String.class, ParameterMode.IN);  
 storedProcedureQuery.registerStoredProcedureParameter("P\_DIR", String.class, ParameterMode.IN);  
 storedProcedureQuery.registerStoredProcedureParameter("P\_FILENAME", String.class, ParameterMode.IN);  
 storedProcedureQuery.registerStoredProcedureParameter("P\_IGNORE\_HEADERLINES", Integer.class,  
 ParameterMode.IN);  
 storedProcedureQuery.registerStoredProcedureParameter("P\_TYPE\_LOAD", Integer.class, ParameterMode.IN);  
 storedProcedureQuery.registerStoredProcedureParameter("P\_ID\_USUARIO", Integer.class, ParameterMode.IN);  
  
  
 // Configuramos el valor de entrada  
 storedProcedureQuery.setParameter("P\_TABLE", table);  
 storedProcedureQuery.setParameter("P\_DIR", dir);  
 storedProcedureQuery.setParameter("P\_FILENAME", filename);  
 storedProcedureQuery.setParameter("P\_IGNORE\_HEADERLINES", ignore);  
 storedProcedureQuery.setParameter("P\_TYPE\_LOAD", typeload);  
 storedProcedureQuery.setParameter("P\_ID\_USUARIO", idUser);  
  
 // Realizamos la llamada al procedimiento  
 storedProcedureQuery.execute();  
  
 // Obtenemos los valores de salida  
 //final String outputValue1 = (String) query.getOutputParameterValue("OUTPUT\_PROCEDURE\_PARAMETER\_NAME1");  
 //final Long outputValue2 = (Long) query.getOutputParameterValue("OUTPUT\_PROCEDURE\_PARAMETER\_NAME2");  
  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
  
 return retorno;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\cargainicial\ICargaInformacion.java

package mx.mbi.dgc.dao.negocio.cargainicial;  
  
import org.springframework.stereotype.Component;  
  
@Component  
public interface ICargaInformacion {  
 public boolean procesaFile(String procedureName,  
 String table,  
 String dir,  
 String filename,  
 Integer ignore,  
 Integer typeload,  
 Integer idUser);  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\cargainicial\package-info.java

package mx.mbi.dgc.dao.negocio.cargainicial;

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DAOAnioMes.java

package mx.mbi.dgc.dao.negocio;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatAnioMes;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOAnioMes extends JpaRepository<CatAnioMes, java.math.BigDecimal> {  
 CatAnioMes findByAnioMesAndCatPlanta(String anioMes, CatPlanta catPlanta) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DAOCargaInformacion.java

package mx.mbi.dgc.dao.negocio;  
  
import mx.mbi.dgc.excepcion.CargaException;  
import mx.mbi.dgc.jpa.entity.CatConexiones;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.jpa.repository.query.Procedure;  
import org.springframework.data.repository.query.Param;  
import org.springframework.scheduling.annotation.Async;  
import org.springframework.stereotype.Repository;  
import org.springframework.transaction.annotation.Transactional;  
  
@Repository  
public interface DAOCargaInformacion extends JpaRepository<CatConexiones, java.math.BigDecimal> {  
 @Async("threadPoolTaskExecutor")  
 @Transactional(timeout = 600)//600 segundos es igual a 10 minutos de espera  
 @Procedure(name = "procesaFileSol")  
 public void procesaFileSol(  
 @Param("P\_TABLE") String pTable,  
 @Param("P\_DIR") String pDir,  
 @Param("P\_FILENAME") String pFileName,  
 @Param("P\_IGNORE\_HEADERLINES") Integer pIgnoreHL,  
 @Param("P\_TYPE\_LOAD") Integer pTypeLoad,  
 @Param("P\_ID\_USUARIO") Integer pIdUsuario  
 ) throws CargaException;  
   
 @Async("threadPoolTaskExecutor")  
 @Transactional(timeout = 600)//600 segundos es igual a 10 minutos de espera  
 @Procedure(name = "procesaFileAguila")  
 public void procesaFileAguila(  
 @Param("P\_TABLE") String pTable,  
 @Param("P\_DIR") String pDir,  
 @Param("P\_FILENAME") String pFileName,  
 @Param("P\_IGNORE\_HEADERLINES") Integer pIgnoreHL,  
 @Param("P\_TYPE\_LOAD") Integer pTypeLoad,  
 @Param("P\_ID\_USUARIO") Integer pIdUsuario  
 ) throws CargaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DAOCargoFijo.java

package mx.mbi.dgc.dao.negocio;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatCargoFijo;  
  
import mx.mbi.dgc.jpa.entity.CatMesCargo;  
  
import mx.mbi.dgc.jpa.entity.CatSubCargos;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOCargoFijo extends JpaRepository<CatCargoFijo, java.math.BigDecimal> {  
 List<CatCargoFijo> findByCatMesCargoAndCatSubCargos(  
 CatMesCargo catMesCargo, CatSubCargos catSubCargos) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DAOCargos.java

package mx.mbi.dgc.dao.negocio;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatCargos;  
  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOCargos extends JpaRepository<CatCargos, java.math.BigDecimal> {  
 List<CatCargos> findByCatPlanta(CatPlanta catPlanta) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DAOCincoVar.java

package mx.mbi.dgc.dao.negocio;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxCincominutalVar;  
  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.data.repository.query.Param;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOCincoVar extends JpaRepository<TxCincominutalVar, java.math.BigDecimal> {  
  
 @Query("select cv FROM TxCincominutalVar cv " +  
 "LEFT JOIN FETCH cv.catVariable catv " +  
 "LEFT JOIN FETCH cv.txCincominutal txc " +  
 "LEFT JOIN FETCH txc.catMesDiaFirma cmdf " +  
 "LEFT JOIN FETCH cmdf.catFirmaCincominutal " +  
 "LEFT JOIN FETCH cmdf.catMesDia ctmd " +  
 "LEFT JOIN FETCH ctmd.catAnioMes cam " +  
 "where " +  
 "cv.txCincominutal.catPlanta.plantaId = :plantaId and " +  
 "cv.estatusDn = 1 and " +  
 "cv.catVariable.variableId = :variableId " +  
 "order by cv.txCincominutal.catMesDiaFirma.mesDiaFirmaId asc")  
 List<TxCincominutalVar> obtenerUltimaValida(@Param("plantaId") BigDecimal plantaId,  
 @Param("variableId") BigDecimal variableId);  
  
  
  
 @Query("select cv FROM TxCincominutalVar cv " +   
 "LEFT JOIN FETCH cv.catVariable catv " +  
 "LEFT JOIN FETCH cv.txCincominutal txc " +   
 "LEFT JOIN FETCH txc.catMesDiaFirma cmdf " +  
 "LEFT JOIN FETCH cmdf.catFirmaCincominutal " +   
 "LEFT JOIN FETCH cmdf.catMesDia ctmd " +  
 "LEFT JOIN FETCH ctmd.catAnioMes cam " +   
 "where " +   
 "cv.txCincominutal.catPlanta.plantaId = :plantaId and " +  
 "cv.estatusDn = 1 " +  
 "order by cv.txCincominutal.catMesDiaFirma.mesDiaFirmaId asc")  
 List<TxCincominutalVar> obtenerPivotes(@Param("plantaId") BigDecimal plantaId);  
   
 @Query("select cv FROM TxCincominutalVar cv " +   
 "LEFT JOIN FETCH cv.catVariable catv " +  
 "LEFT JOIN FETCH cv.txCincominutal txc " +   
 "LEFT JOIN FETCH txc.catMesDiaFirma cmdf " +  
 "LEFT JOIN FETCH cmdf.catFirmaCincominutal " +   
 "LEFT JOIN FETCH cmdf.catMesDia ctmd " +  
 "LEFT JOIN FETCH ctmd.catAnioMes cam " +   
 "where " +   
 "cv.txCincominutal.catPlanta.plantaId = :plantaId and " +  
 "cv.estatusDn = 1 and " +  
 "cv.catVariable.variableId = :variableId and " +  
 "cv.cincominutalVarId > :idInicio and " +  
 "cv.cincominutalVarId < :idFinal " +  
 "order by cv.txCincominutal.catMesDiaFirma.mesDiaFirmaId asc")  
 List<TxCincominutalVar> obtenerTxCincominutalVarEntre(  
 @Param("plantaId") BigDecimal plantaId,  
 @Param("variableId") BigDecimal variableId,  
 @Param("idInicio") BigDecimal idInicio,  
 @Param("idFinal") BigDecimal idFinal);  
  
  
 @Query("select cv FROM TxCincominutalVar cv " +  
 "LEFT JOIN FETCH cv.catVariable catv " +  
 "LEFT JOIN FETCH cv.txCincominutal txc " +  
 "LEFT JOIN FETCH txc.catMesDiaFirma cmdf " +  
 "LEFT JOIN FETCH cmdf.catFirmaCincominutal " +  
 "LEFT JOIN FETCH cmdf.catMesDia ctmd " +  
 "LEFT JOIN FETCH ctmd.catAnioMes cam " +  
 "where " +  
 "cv.txCincominutal.catPlanta.plantaId = :plantaId and " +  
 "cv.catVariable.variableId = :variableId and " +  
 "cmdf.mesDiaFirmaId >= :idMesDiaFirmaInicio and " +  
 "cmdf.mesDiaFirmaId <= :idMesDiaFirmaFinal " +  
 "order by cv.txCincominutal.catMesDiaFirma.mesDiaFirmaId asc")  
 List<TxCincominutalVar> obtenerTxCincominutalVarEntreCincominutales(  
 @Param("plantaId") BigDecimal plantaId,  
 @Param("variableId") BigDecimal variableId,  
 @Param("idMesDiaFirmaInicio") BigDecimal idMesDiaFirmaInicio,  
 @Param("idMesDiaFirmaFinal") BigDecimal idMesDiaFirmaFinal);  
  
  
 @Query("select MIN(cv.cincominutalVarId) FROM TxCincominutalVar cv " +  
 "where " +  
 "cv.txCincominutal.catPlanta.plantaId = :plantaId and " +  
 "cv.catVariable.variableId = :variableId " +   
 "and cv.estatusDn = :estatusDn " /\*+  
 "and cv.estatusCn = :estatusCn"\*/)  
 BigDecimal getTxCincominutalVarPrimera(@Param("variableId") BigDecimal variableId,  
 @Param("estatusDn") BigDecimal estatusDn,  
 //@Param("estatusCn") BigDecimal estatusCn,  
 @Param("plantaId") BigDecimal plantaId);  
  
  
 @Query("select cv FROM TxCincominutalVar cv " + "LEFT JOIN FETCH cv.catVariable catv " +  
 "LEFT JOIN FETCH cv.txCincominutal txc " + "LEFT JOIN FETCH txc.catMesDiaFirma cmdf " +  
 "LEFT JOIN FETCH cmdf.catFirmaCincominutal " + "LEFT JOIN FETCH cmdf.catMesDia ctmd " +  
 "LEFT JOIN FETCH ctmd.catAnioMes cam " +   
 "where " +   
 "cv.txCincominutal.catPlanta.plantaId = :plantaId and " +  
 "cv.estatusDn = :estatusDn " + "and cv.catVariable.variableId = :variableId " +  
 "order by cv.txCincominutal.catMesDiaFirma.mesDiaFirmaId asc")  
 List<TxCincominutalVar> getListaVars(@Param("estatusDn") BigDecimal estatusDn,  
 @Param("variableId") BigDecimal variableId,  
 @Param("plantaId") BigDecimal plantaId) throws DyCNormaException;  
  
 @Query("select txVar from TxCincominutalVar txVar " + "LEFT JOIN FETCH txVar.catVariable catv " +  
 "LEFT JOIN FETCH txVar.txCincominutal txc " + "LEFT JOIN FETCH txc.catMesDiaFirma cmdf " +  
 "LEFT JOIN FETCH cmdf.catFirmaCincominutal " + "LEFT JOIN FETCH cmdf.catMesDia ctmd " +  
 "LEFT JOIN FETCH ctmd.catAnioMes cam " +   
 "where " +  
 "txVar.txCincominutal.catPlanta.plantaId = :plantaId and " +  
 "txVar.estatusDn = :estatusDn " +  
 "and txVar.estatusCn = :estatusCn " + "and txVar.catVariable.variableId = :variableId " +  
 "order by txVar.txCincominutal.catMesDiaFirma.mesDiaFirmaId asc")  
 List<TxCincominutalVar> getPrimeraValidaCorreccion(@Param("estatusDn") BigDecimal estatusDn,  
 @Param("estatusCn") BigDecimal estatusCn,  
 @Param("variableId") BigDecimal variableId,  
 @Param("plantaId") BigDecimal plantaId,  
 Pageable pageable) throws DyCNormaException;  
  
 @Query("select txVar from TxCincominutalVar txVar " + "LEFT JOIN FETCH txVar.catVariable catv " +  
 "LEFT JOIN FETCH txVar.txCincominutal txc " + "LEFT JOIN FETCH txc.catMesDiaFirma cmdf " +  
 "LEFT JOIN FETCH cmdf.catFirmaCincominutal " + "LEFT JOIN FETCH cmdf.catMesDia ctmd " +  
 "LEFT JOIN FETCH ctmd.catAnioMes cam " +   
 "where " +   
 "txVar.txCincominutal.catPlanta.plantaId = :plantaId " +  
 "and txVar.estatusCn = :estatusCn " +  
 "and txVar.catVariable.variableId = :variableId " + "and txc.cincominutalId >= :cincominutalId " +  
 "order by txc.catMesDiaFirma.mesDiaFirmaId asc")  
 List<TxCincominutalVar> getListaAmbientales(@Param("estatusCn") BigDecimal estatusCn,  
 @Param("variableId") BigDecimal variableId,  
 @Param("cincominutalId") BigDecimal cincominutalId,  
 @Param("plantaId") BigDecimal plantaId);  
  
  
 @Query("select cv FROM TxCincominutalVar cv " + "LEFT JOIN FETCH cv.catVariable catv " +  
 "LEFT JOIN FETCH cv.txCincominutal txc " + "LEFT JOIN FETCH txc.catMesDiaFirma cmdf " +  
 "LEFT JOIN FETCH cmdf.catFirmaCincominutal " + "LEFT JOIN FETCH cmdf.catMesDia ctmd " +  
 "LEFT JOIN FETCH ctmd.catAnioMes cam " +   
 "where " +  
 "cv.txCincominutal.catPlanta.plantaId = :plantaId and " +  
 "cv.txCincominutal.catMesDiaFirma.mesDiaFirmaId = :mesDiaFirmaId " +  
 "and cv.catVariable.variableId = :variableId")  
 TxCincominutalVar getTxCincoVarAnterior(@Param("mesDiaFirmaId") BigDecimal mesDiaFirmaId,  
 @Param("variableId") BigDecimal variableId,  
 @Param("plantaId") BigDecimal plantaId) throws DyCNormaException;  
  
 @Query("select cv FROM TxCincominutalVar cv " + "LEFT JOIN FETCH cv.catVariable catv " +  
 "LEFT JOIN FETCH cv.txCincominutal txc " + "LEFT JOIN FETCH txc.catMesDiaFirma cmdf " + "LEFT JOIN FETCH cmdf.catFirmaCincominutal " +  
 "LEFT JOIN FETCH cmdf.catMesDia ctmd " + "LEFT JOIN FETCH ctmd.catAnioMes cam " +  
 "where " +  
 "cv.txCincominutal.catMesDiaFirma.mesDiaFirmaId = " +  
 ":mesDiaFirmaId and cv.catVariable.variableId = :variableId " +  
 "and cv.txCincominutal.catPlanta.plantaId = :plantaId")  
 TxCincominutalVar getTxCincoVarOtraPlanta(@Param("mesDiaFirmaId") BigDecimal mesDiaFirmaId,  
 @Param("variableId") BigDecimal variableId,  
 @Param("plantaId") BigDecimal plantaId) throws DyCNormaException;  
  
 @Query("select MIN(cv.txCincominutal.cincominutalId) FROM TxCincominutalVar cv where " +  
 "cv.estatusDn = :estatusDn " +   
 "and cv.catVariable.tipo = :tipo " +  
 "and cv.txCincominutal.catPlanta.plantaId = :plantaId " +   
 "and cv.estatusCn = :estatusCn ")  
 BigDecimal getCincoPrimerValido(@Param("estatusDn") BigDecimal estatusDn, @Param("tipo") String tipo,  
 @Param("plantaId") BigDecimal plantaId, @Param("estatusCn") BigDecimal estatusCn);  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DAODyCNorma.java

package mx.mbi.dgc.dao.negocio;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatMesDiaFirma;  
import mx.mbi.dgc.jpa.entity.TxCincominutal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.data.repository.query.Param;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAODyCNorma extends JpaRepository<TxCincominutal, BigDecimal> {  
 TxCincominutal findByCatMesDiaFirma(CatMesDiaFirma catMesDiaFirma) throws DyCNormaException;  
  
 @Query("select cinco FROM TxCincominutal cinco " +  
 "RIGHT JOIN FETCH cinco.txCincominutalVars txcv " +  
 "INNER JOIN FETCH txcv.catVariable catv " +  
 "LEFT JOIN FETCH cinco.catMesDiaFirma cmdf " +  
 "LEFT JOIN FETCH cmdf.catFirmaCincominutal " +  
 "LEFT JOIN FETCH cmdf.catMesDia ctmd " +  
 "LEFT JOIN FETCH ctmd.catAnioMes cam " +  
 "where " +  
 "txcv.estatusCn = :estatusCn and " +  
 "cinco.cincominutalId > :cincominutalId and " +  
 "cinco.catPlanta.plantaId = :plantaId and " +  
 "catv.tipo = :tipo " +  
 "order by cinco.catMesDiaFirma.mesDiaFirmaId asc")  
 List<TxCincominutal> getCincominutalesCorregirCroma(  
 @Param("cincominutalId") BigDecimal cincominutalId,  
 @Param("plantaId") BigDecimal plantaId,  
 @Param("estatusCn") BigDecimal estatusCn,  
 @Param("tipo") String tipo  
 );  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DAOEscenarioVariable.java

package mx.mbi.dgc.dao.negocio;  
  
import mx.mbi.dgc.jpa.entity.TxEscenarioVariable;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOEscenarioVariable extends JpaRepository<TxEscenarioVariable, java.math.BigDecimal> {  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DAOLogin.java

package mx.mbi.dgc.dao.negocio;  
  
import mx.mbi.dgc.jpa.entity.TxUsuario;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOLogin extends JpaRepository<TxUsuario, java.math.BigDecimal> {  
 TxUsuario findByUsername(String username) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DAOMesDia.java

package mx.mbi.dgc.dao.negocio;  
  
import java.math.BigDecimal;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatAnioMes;  
  
import mx.mbi.dgc.jpa.entity.CatMesDia;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOMesDia extends JpaRepository<CatMesDia, java.math.BigDecimal> {  
 CatMesDia findByCatAnioMesAndDia(CatAnioMes catAnioMes, BigDecimal dia) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DAOMesDiaFirma.java

package mx.mbi.dgc.dao.negocio;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatMesDia;  
import mx.mbi.dgc.jpa.entity.CatMesDiaFirma;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.data.repository.query.Param;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOMesDiaFirma extends JpaRepository<CatMesDiaFirma, java.math.BigDecimal> {  
 List<CatMesDiaFirma> findAllByCatMesDia(CatMesDia catMesDia) throws DyCNormaException;  
   
// select cfc.FIRMA, cam.ANIO\_MES, cmd.DIA from TX\_CINCOMINUTAL c  
// inner join CAT\_PLANTA p on p.PLANTA\_ID = c.PLANTA\_ID  
// inner join CAT\_MES\_DIA\_FIRMA cmdf on cmdf.MES\_DIA\_FIRMA\_ID = c.MES\_DIA\_FIRMA\_ID   
// inner join CAT\_FIRMA\_CINCOMINUTAL cfc on cfc.FIRMA\_ID = cmdf.FIRMA\_ID  
// inner join CAT\_MES\_DIA cmd on cmd.MES\_DIA\_ID = cmdf.MES\_DIA\_ID  
// inner join CAT\_ANIO\_MES cam on cam.ANIO\_MES\_ID = cmd.ANIO\_MES\_ID  
// where   
// cmd.DIA = 31 and  
// cam.ANIO\_MES = '2018\_08' and  
// cfc.FIRMA = '23502355' and   
// p.PLANTA\_ID = 2  
// order by p.PLANTA\_ID, cam.ANIO\_MES, cmd.DIA, cfc.FIRMA desc  
// ;  
 @Query("SELECT cmdf FROM CatMesDiaFirma cmdf where " +  
 "cmdf.catMesDia.dia = :dia " +  
 "and cmdf.catMesDia.catAnioMes.anioMes = :anioMes " +  
 "and cmdf.catFirmaCincominutal.firma = :firma " +  
 "and cmdf.catMesDia.catAnioMes.catPlanta.plantaId = :plantaId")  
 CatMesDiaFirma getCatMesDiaFirma(  
 @Param("dia")BigDecimal dia,   
 @Param("anioMes")String anioMes,   
 @Param("firma")String firma,   
 @Param("plantaId")BigDecimal plantaId);  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DaoNegocioCatCondiciones.java

package mx.mbi.dgc.dao.negocio;  
  
import java.util.List;  
  
import mx.mbi.dgc.jpa.entity.CatCondiciones;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DaoNegocioCatCondiciones extends JpaRepository<CatCondiciones, java.math.BigDecimal> {  
 List<CatCondiciones> findAllByProcesoAndCatPlanta(String proceso, CatPlanta catPlanta);  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DaoNegocioCatFirmaCincominutal.java

package mx.mbi.dgc.dao.negocio;  
  
import java.math.BigDecimal;  
  
import mx.mbi.dgc.jpa.entity.CatFirmaCincominutal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.data.repository.query.Param;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DaoNegocioCatFirmaCincominutal extends JpaRepository<CatFirmaCincominutal, BigDecimal> {  
 @Query("SELECT catFirmaCincominutal FROM CatFirmaCincominutal catFirmaCincominutal where " +  
 "catFirmaCincominutal.firma = :firma")  
 CatFirmaCincominutal getCatFirmaCincominutal (@Param("firma")String firma);  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DaoNegocioCatMedidores.java

package mx.mbi.dgc.dao.negocio;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatMedidores;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DaoNegocioCatMedidores extends JpaRepository<CatMedidores, BigDecimal> {  
 CatMedidores findByNombreAndCatPlanta(String nombre, CatPlanta catPlanta) throws DyCNormaException;  
 List<CatMedidores> findAllByCatPlanta(CatPlanta catPlanta) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DaoNegocioCatVariablePerfil.java

package mx.mbi.dgc.dao.negocio;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxCincominutal;  
  
import mx.mbi.dgc.jpa.entity.TxPerfilesVariable;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DaoNegocioCatVariablePerfil extends JpaRepository<TxPerfilesVariable, BigDecimal> {  
 List<TxPerfilesVariable> findAllByTxCincominutal(TxCincominutal txCincominutal) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DaoNegocioTxCincominutal.java

package mx.mbi.dgc.dao.negocio;  
  
import java.math.BigDecimal;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatMesDiaFirma;  
import mx.mbi.dgc.jpa.entity.TxCincominutal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DaoNegocioTxCincominutal extends JpaRepository<TxCincominutal, BigDecimal> {  
 TxCincominutal findByCatMesDiaFirma(CatMesDiaFirma catMesDiaFirma) throws DyCNormaException;  
   
 /\*@Query("SELECT txCincominutal FROM TxCincominutal txCincominutal where " +  
 "txCincominutal.catMesDiaFirma.catMesDia.catAnioMes = :catAnioMes and" +  
 "txCincominutal.catPlanta = :catPlanta")  
 List<TxCincominutal> getTxCincominutalAnioMes(CatAnioMes catAnioMes, CatPlanta catPlanta);\*/  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DaoNegocioTxCincominutalVar.java

package mx.mbi.dgc.dao.negocio;  
  
import java.math.BigDecimal;  
  
import java.util.Date;  
import java.util.List;  
  
import mx.mbi.dgc.jpa.entity.TxCincominutalVar;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.data.repository.query.Param;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DaoNegocioTxCincominutalVar extends JpaRepository<TxCincominutalVar, BigDecimal> {  
   
 @Query("Select txcv from TxCincominutalVar txcv where " +  
 "txcv.txCincominutal.catMesDiaFirma.fechaFirma >= :fechaInicio " +   
 "and txcv.txCincominutal.catMesDiaFirma.fechaFirma <= :fechaFinal " +  
 "and txcv.txCincominutal.catPlanta.plantaId = :plantaId " +  
 "order by FUNCTION('To\_Char',txcv.txCincominutal.catMesDiaFirma.fechaFirma,'YYYY/MM/DD' ) asc," + "txcv.txCincominutal.catMesDiaFirma.catFirmaCincominutal.firmaId asc")  
 List<TxCincominutalVar> getTxCincominutalVarByRango(@Param("fechaInicio")Date fechaInicio, @Param("fechaFinal") Date fechaFinal,  
 @Param("plantaId") BigDecimal plantaId);  
  
   
 @Query("Select txcv from TxCincominutalVar txcv where " +  
 "txcv.txCincominutal.catMesDiaFirma.fechaFirma >= :fechaInicio " +   
 "and txcv.txCincominutal.catMesDiaFirma.fechaFirma <= :fechaFinal")  
 List<TxCincominutalVar> getTxCincominutalVarByRango(@Param("fechaInicio")Date fechaInicio, @Param("fechaFinal") Date fechaFinal);  
  
 @Query("Select txcv from TxCincominutalVar txcv " + "LEFT JOIN FETCH txcv.txCincominutal txc " +  
 "LEFT JOIN FETCH txc.catMesDiaFirma cmdf " + "LEFT JOIN FETCH cmdf.catFirmaCincominutal " +  
 "LEFT JOIN FETCH cmdf.catMesDia ctmd " + "LEFT JOIN FETCH ctmd.catAnioMes cam " +   
 "where " +  
 "txcv.txCincominutal.cincominutalId = :cincominutalId")  
 List<TxCincominutalVar> getTxCincominutalVarByTxCincominutal(@Param("cincominutalId") BigDecimal cincominutalId);  
  
  
 }  
   
/\* @Query("SELECT \n" +   
 " TEMP\_BULBO.fecha\_firma,\n" +   
 " TEMP\_BULBO.\"TEMP DE BULBO SECO VALOR ORIGINAL\",\n" +   
 " TEMP\_BULBO.\"TEMP DE BULBO SECO VALOR FORMATO\",\n" +   
 " TEMP\_BULBO.estatus\_df ,\n" +   
 " TEMP\_BULBO.estatus\_cf,\n" +   
 " TEMP\_BULBO.estatus\_dn,\n" +   
 " TEMP\_BULBO.estatus\_cn,\n" +   
 " TEMP\_BULBO.valor\_norma,\n" +   
 " HUMEDAD\_RE.\"HUMEDAD RELATIVA VALOR ORIGINAL\",\n" +   
 " HUMEDAD\_RE.\"HUMEDAD RELATIVA VALOR FORMATO\",\n" +   
 " HUMEDAD\_RE.estatus\_df ,\n" +   
 " HUMEDAD\_RE.estatus\_cf,\n" +   
 " HUMEDAD\_RE.estatus\_dn,\n" +   
 " HUMEDAD\_RE.estatus\_cn,\n" +   
 " HUMEDAD\_RE.valor\_norma,\n" +   
 " PRESION\_CONDENSADOR.\"PRESION DEL CONDENSADOR VALOR ORIGINAL\",\n" +   
 " PRESION\_CONDENSADOR.\"PRESION DEL CONDENSADOR VALOR FORMATO\",\n" +   
 " PRESION\_CONDENSADOR.estatus\_df ,\n" +   
 " PRESION\_CONDENSADOR.estatus\_cf,\n" +   
 " PRESION\_CONDENSADOR.estatus\_dn,\n" +   
 " PRESION\_CONDENSADOR.estatus\_cn,\n" +   
 " PRESION\_CONDENSADOR.valor\_norma,\n" +   
 " PRESION\_ATMOSFERICA.\"PRESION ATMOSFERICA VALOR ORIGINAL\",\n" +   
 " PRESION\_ATMOSFERICA.\"PRESION ATMOSFERICA VALOR FORMATO\",\n" +   
 " PRESION\_ATMOSFERICA.estatus\_df ,\n" +   
 " PRESION\_ATMOSFERICA.estatus\_cf,\n" +   
 " PRESION\_ATMOSFERICA.estatus\_dn,\n" +   
 " PRESION\_ATMOSFERICA.estatus\_cn,\n" +   
 " PRESION\_ATMOSFERICA.valor\_norma,\n" +   
 " PODER\_CALORIFICO\_IM.\"PODER CALORIFICO INFERIOR MASICO VALOR ORIGINAL\",\n" +   
 " PODER\_CALORIFICO\_IM.\"PODER CALORIFICO INFERIOR MASICO VALOR FORMATO\",\n" +   
 " PODER\_CALORIFICO\_IM.estatus\_df ,\n" +   
 " PODER\_CALORIFICO\_IM.estatus\_cf,\n" +   
 " PODER\_CALORIFICO\_IM.estatus\_dn,\n" +   
 " PODER\_CALORIFICO\_IM.estatus\_cn,\n" +   
 " PODER\_CALORIFICO\_IM.valor\_norma,\n" +   
 " PODER\_CALORIFICO\_SM.\"PODER CALORIFICO SUPERIOR MASICO VALOR ORIGINAL\",\n" +   
 " PODER\_CALORIFICO\_SM.\"PODER CALORIFICO SUPERIOR MASICO VALOR FORMATO\",\n" +   
 " PODER\_CALORIFICO\_SM.estatus\_df ,\n" +   
 " PODER\_CALORIFICO\_SM.estatus\_cf,\n" +   
 " PODER\_CALORIFICO\_SM.estatus\_dn,\n" +   
 " PODER\_CALORIFICO\_SM.estatus\_cn,\n" +   
 " PODER\_CALORIFICO\_SM.valor\_norma,\n" +   
 " PODER\_CALORIFICO\_IV.\"PODER CALORIFICO INFERIOR VOLUMETRICO VALOR ORIGINAL\",\n" +   
 " PODER\_CALORIFICO\_IV.\"PODER CALORIFICO INFERIOR VOLUMETRICO VALOR FORMATO\",\n" +   
 " PODER\_CALORIFICO\_IV.estatus\_df ,\n" +   
 " PODER\_CALORIFICO\_IV.estatus\_cf,\n" +   
 " PODER\_CALORIFICO\_IV.estatus\_dn,\n" +   
 " PODER\_CALORIFICO\_IV.estatus\_cn,\n" +   
 " PODER\_CALORIFICO\_IV.valor\_norma,\n" +   
 " PODER\_CALORIFICO\_SV.\"PODER CALORIFICO SUPERIOR VOLUMETRICO VALOR ORIGINAL\",\n" +   
 " PODER\_CALORIFICO\_SV.\"PODER CALORIFICO SUPERIOR VOLUMETRICO VALOR FORMATO\",\n" +   
 " PODER\_CALORIFICO\_SV.estatus\_df ,\n" +   
 " PODER\_CALORIFICO\_SV.estatus\_cf,\n" +   
 " PODER\_CALORIFICO\_SV.estatus\_dn,\n" +   
 " PODER\_CALORIFICO\_SV.estatus\_cn,\n" +   
 " PODER\_CALORIFICO\_SV.valor\_norma\n" +   
 "FROM (\n" +   
 "SELECT\n" +   
 " dgc2.cat\_mes\_dia\_firma.fecha\_firma,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_original \"TEMP DE BULBO SECO VALOR ORIGINAL\",\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_formato \"TEMP DE BULBO SECO VALOR FORMATO\",\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_df ,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cf,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_dn,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cn,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_norma,\n" +   
 " dgc2.tx\_cincominutal.planta\_id\n" +   
 " FROM\n" +   
 " dgc2.tx\_cincominutal\_var\n" +   
 " INNER JOIN dgc2.tx\_cincominutal ON dgc2.tx\_cincominutal\_var.cincominutal\_id = dgc2.tx\_cincominutal.cincominutal\_id\n" +   
 " INNER JOIN dgc2.cat\_mes\_dia\_firma ON dgc2.cat\_mes\_dia\_firma.mes\_dia\_firma\_id = dgc2.tx\_cincominutal.mes\_dia\_firma\_id\n" +   
 " INNER JOIN dgc2.cat\_variable ON dgc2.cat\_variable.variable\_id = dgc2.tx\_cincominutal\_var.variable\_id\n" +   
 " WHERE\n" +   
 " dgc2.tx\_cincominutal.planta\_id = 1\n" +   
 " AND dgc2.cat\_variable.variable\_id = 14) TEMP\_BULBO,\n" +   
 " (\n" +   
 "SELECT\n" +   
 " dgc2.cat\_mes\_dia\_firma.fecha\_firma,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_original \"HUMEDAD RELATIVA VALOR ORIGINAL\",\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_formato \"HUMEDAD RELATIVA VALOR FORMATO\",\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_df ,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cf,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_dn,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cn,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_norma,\n" +   
 " dgc2.tx\_cincominutal.planta\_id\n" +   
 " FROM\n" +   
 " dgc2.tx\_cincominutal\_var\n" +   
 " INNER JOIN dgc2.tx\_cincominutal ON dgc2.tx\_cincominutal\_var.cincominutal\_id = dgc2.tx\_cincominutal.cincominutal\_id\n" +   
 " INNER JOIN dgc2.cat\_mes\_dia\_firma ON dgc2.cat\_mes\_dia\_firma.mes\_dia\_firma\_id = dgc2.tx\_cincominutal.mes\_dia\_firma\_id\n" +   
 " INNER JOIN dgc2.cat\_variable ON dgc2.cat\_variable.variable\_id = dgc2.tx\_cincominutal\_var.variable\_id\n" +   
 " WHERE\n" +   
 " dgc2.tx\_cincominutal.planta\_id = 1\n" +   
 " AND dgc2.cat\_variable.variable\_id = 8) HUMEDAD\_RE,\n" +   
 " (\n" +   
 "SELECT\n" +   
 " dgc2.cat\_mes\_dia\_firma.fecha\_firma,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_original \"PRESION DEL CONDENSADOR VALOR ORIGINAL\",\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_formato \"PRESION DEL CONDENSADOR VALOR FORMATO\",\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_df ,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cf,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_dn,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cn,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_norma,\n" +   
 " dgc2.tx\_cincominutal.planta\_id\n" +   
 " FROM\n" +   
 " dgc2.tx\_cincominutal\_var\n" +   
 " INNER JOIN dgc2.tx\_cincominutal ON dgc2.tx\_cincominutal\_var.cincominutal\_id = dgc2.tx\_cincominutal.cincominutal\_id\n" +   
 " INNER JOIN dgc2.cat\_mes\_dia\_firma ON dgc2.cat\_mes\_dia\_firma.mes\_dia\_firma\_id = dgc2.tx\_cincominutal.mes\_dia\_firma\_id\n" +   
 " INNER JOIN dgc2.cat\_variable ON dgc2.cat\_variable.variable\_id = dgc2.tx\_cincominutal\_var.variable\_id\n" +   
 " WHERE\n" +   
 " dgc2.tx\_cincominutal.planta\_id = 1\n" +   
 " AND dgc2.cat\_variable.variable\_id = 9) PRESION\_CONDENSADOR,\n" +   
 " (\n" +   
 "SELECT\n" +   
 " dgc2.cat\_mes\_dia\_firma.fecha\_firma,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_original \"PRESION ATMOSFERICA VALOR ORIGINAL\",\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_formato \"PRESION ATMOSFERICA VALOR FORMATO\",\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_df ,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cf,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_dn,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cn,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_norma,\n" +   
 " dgc2.tx\_cincominutal.planta\_id\n" +   
 " FROM\n" +   
 " dgc2.tx\_cincominutal\_var\n" +   
 " INNER JOIN dgc2.tx\_cincominutal ON dgc2.tx\_cincominutal\_var.cincominutal\_id = dgc2.tx\_cincominutal.cincominutal\_id\n" +   
 " INNER JOIN dgc2.cat\_mes\_dia\_firma ON dgc2.cat\_mes\_dia\_firma.mes\_dia\_firma\_id = dgc2.tx\_cincominutal.mes\_dia\_firma\_id\n" +   
 " INNER JOIN dgc2.cat\_variable ON dgc2.cat\_variable.variable\_id = dgc2.tx\_cincominutal\_var.variable\_id\n" +   
 " WHERE\n" +   
 " dgc2.tx\_cincominutal.planta\_id = 1\n" +   
 " AND dgc2.cat\_variable.variable\_id = 10) PRESION\_ATMOSFERICA,\n" +   
 " (\n" +   
 "SELECT\n" +   
 " dgc2.cat\_mes\_dia\_firma.fecha\_firma,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_original \"PODER CALORIFICO INFERIOR MASICO VALOR ORIGINAL\",\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_formato \"PODER CALORIFICO INFERIOR MASICO VALOR FORMATO\",\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_df ,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cf,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_dn,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cn,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_norma,\n" +   
 " dgc2.tx\_cincominutal.planta\_id\n" +   
 " FROM\n" +   
 " dgc2.tx\_cincominutal\_var\n" +   
 " INNER JOIN dgc2.tx\_cincominutal ON dgc2.tx\_cincominutal\_var.cincominutal\_id = dgc2.tx\_cincominutal.cincominutal\_id\n" +   
 " INNER JOIN dgc2.cat\_mes\_dia\_firma ON dgc2.cat\_mes\_dia\_firma.mes\_dia\_firma\_id = dgc2.tx\_cincominutal.mes\_dia\_firma\_id\n" +   
 " INNER JOIN dgc2.cat\_variable ON dgc2.cat\_variable.variable\_id = dgc2.tx\_cincominutal\_var.variable\_id\n" +   
 " WHERE\n" +   
 " dgc2.tx\_cincominutal.planta\_id = 1\n" +   
 " AND dgc2.cat\_variable.variable\_id = 11) PODER\_CALORIFICO\_IM,\n" +   
 " (\n" +   
 "SELECT\n" +   
 " dgc2.cat\_mes\_dia\_firma.fecha\_firma,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_original \"PODER CALORIFICO SUPERIOR MASICO VALOR ORIGINAL\",\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_formato \"PODER CALORIFICO SUPERIOR MASICO VALOR FORMATO\",\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_df ,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cf,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_dn,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cn,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_norma,\n" +   
 " dgc2.tx\_cincominutal.planta\_id\n" +   
 " FROM\n" +   
 " dgc2.tx\_cincominutal\_var\n" +   
 " INNER JOIN dgc2.tx\_cincominutal ON dgc2.tx\_cincominutal\_var.cincominutal\_id = dgc2.tx\_cincominutal.cincominutal\_id\n" +   
 " INNER JOIN dgc2.cat\_mes\_dia\_firma ON dgc2.cat\_mes\_dia\_firma.mes\_dia\_firma\_id = dgc2.tx\_cincominutal.mes\_dia\_firma\_id\n" +   
 " INNER JOIN dgc2.cat\_variable ON dgc2.cat\_variable.variable\_id = dgc2.tx\_cincominutal\_var.variable\_id\n" +   
 " WHERE\n" +   
 " dgc2.tx\_cincominutal.planta\_id = 1\n" +   
 " AND dgc2.cat\_variable.variable\_id = 13) PODER\_CALORIFICO\_SM,\n" +   
 " (\n" +   
 "SELECT\n" +   
 " dgc2.cat\_mes\_dia\_firma.fecha\_firma,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_original \"PODER CALORIFICO INFERIOR VOLUMETRICO VALOR ORIGINAL\",\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_formato \"PODER CALORIFICO INFERIOR VOLUMETRICO VALOR FORMATO\",\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_df ,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cf,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_dn,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cn,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_norma,\n" +   
 " dgc2.tx\_cincominutal.planta\_id\n" +   
 " FROM\n" +   
 " dgc2.tx\_cincominutal\_var\n" +   
 " INNER JOIN dgc2.tx\_cincominutal ON dgc2.tx\_cincominutal\_var.cincominutal\_id = dgc2.tx\_cincominutal.cincominutal\_id\n" +   
 " INNER JOIN dgc2.cat\_mes\_dia\_firma ON dgc2.cat\_mes\_dia\_firma.mes\_dia\_firma\_id = dgc2.tx\_cincominutal.mes\_dia\_firma\_id\n" +   
 " INNER JOIN dgc2.cat\_variable ON dgc2.cat\_variable.variable\_id = dgc2.tx\_cincominutal\_var.variable\_id\n" +   
 " WHERE\n" +   
 " dgc2.tx\_cincominutal.planta\_id = 1\n" +   
 " AND dgc2.cat\_variable.variable\_id = 15) PODER\_CALORIFICO\_IV,\n" +   
 " (\n" +   
 "SELECT\n" +   
 " dgc2.cat\_mes\_dia\_firma.fecha\_firma,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_original \"PODER CALORIFICO SUPERIOR VOLUMETRICO VALOR ORIGINAL\",\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_formato \"PODER CALORIFICO SUPERIOR VOLUMETRICO VALOR FORMATO\",\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_df ,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cf,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_dn,\n" +   
 " dgc2.tx\_cincominutal\_var.estatus\_cn,\n" +   
 " dgc2.tx\_cincominutal\_var.valor\_norma,\n" +   
 " dgc2.tx\_cincominutal.planta\_id\n" +   
 " FROM\n" +   
 " dgc2.tx\_cincominutal\_var\n" +   
 " INNER JOIN dgc2.tx\_cincominutal ON dgc2.tx\_cincominutal\_var.cincominutal\_id = dgc2.tx\_cincominutal.cincominutal\_id\n" +   
 " INNER JOIN dgc2.cat\_mes\_dia\_firma ON dgc2.cat\_mes\_dia\_firma.mes\_dia\_firma\_id = dgc2.tx\_cincominutal.mes\_dia\_firma\_id\n" +   
 " INNER JOIN dgc2.cat\_variable ON dgc2.cat\_variable.variable\_id = dgc2.tx\_cincominutal\_var.variable\_id\n" +   
 " WHERE\n" +   
 " dgc2.tx\_cincominutal.planta\_id = 1\n" +   
 " AND dgc2.cat\_variable.variable\_id = 16) PODER\_CALORIFICO\_SV\n" +   
 " WHERE TEMP\_BULBO.fecha\_firma = HUMEDAD\_RE.fecha\_firma\n" +   
 " AND HUMEDAD\_RE.fecha\_firma = PRESION\_CONDENSADOR.fecha\_firma\n" +   
 " AND PRESION\_CONDENSADOR.fecha\_firma = PRESION\_ATMOSFERICA.fecha\_firma\n" +   
 " AND PRESION\_ATMOSFERICA.fecha\_firma = PODER\_CALORIFICO\_IM.fecha\_firma\n" +   
 " AND PODER\_CALORIFICO\_IM.fecha\_firma = PODER\_CALORIFICO\_SM.fecha\_firma\n" +   
 " AND PODER\_CALORIFICO\_SM.fecha\_firma = PODER\_CALORIFICO\_IV.fecha\_firma\n" +   
 " AND PODER\_CALORIFICO\_IV.fecha\_firma = PODER\_CALORIFICO\_SV.fecha\_firma\n" +   
 " ORDER BY TEMP\_BULBO.fecha\_firma ASC;\n" +   
 "and txcv.txCincominutal.catMesDiaFirma.fechaFirma <= :fechaFinal " +  
 "and txcv.txCincominutal.catPlanta.plantaId = :plantaId " +  
 "order by txcv.txCincominutal.catMesDiaFirma.fechaFirma")  
 List<TxCincominutalVar> getTxCincominutalVarByRango(@Param("fechaInicio")Date fechaInicio, @Param("fechaFinal") Date fechaFinal,  
 @Param("plantaId") BigDecimal plantaId);\*/  
   
 /\* @Query("Select txcv from TxCincominutalVar txcv")  
 List<TxCincominutalVar> getTxCincominutalVarByRango();\*/  
   
 /\*@Query(value = "Select \* from Tx\_Cincominutal\_Var where FECHA\_CARGA >= :fechaInicio and FECHA\_CARGA <= :fechaFinal", nativeQuery=true)  
 List<TxCincominutalVar> getTxCincominutalVarByRango(@Param("fechaInicio")Date fechaInicio, @Param("fechaFinal") Date fechaFinal);\*/

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DaoNegocioTxPerfilesVariable.java

package mx.mbi.dgc.dao.negocio;  
  
import java.math.BigDecimal;  
  
import java.util.Date;  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatMedidores;  
import mx.mbi.dgc.jpa.entity.CatVariablePerfil;  
import mx.mbi.dgc.jpa.entity.TxCincominutal;  
import mx.mbi.dgc.jpa.entity.TxPerfilesVariable;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.data.repository.query.Param;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DaoNegocioTxPerfilesVariable extends JpaRepository<TxPerfilesVariable, BigDecimal> {  
 TxPerfilesVariable findByTxCincominutalAndCatMedidoresAndCatVariablePerfilAndPrincipal(TxCincominutal txCincominutal,  
 CatMedidores catMedidores,  
 CatVariablePerfil catVariablePerfil,  
 BigDecimal principal) throws DyCNormaException;  
  
 List<TxPerfilesVariable> findByTxCincominutalAndCatMedidoresAndPrincipal(TxCincominutal txCincominutal,  
 CatMedidores catMedidores,  
 BigDecimal principal) throws DyCNormaException;  
  
 @Query("Select txpv from TxPerfilesVariable txpv where " +  
 "txpv.txCincominutal.catMesDiaFirma.fechaFirma >= :fechaInicio " +  
 "and txpv.txCincominutal.catMesDiaFirma.fechaFirma <= :fechaFinal " +  
 "and txpv.txCincominutal.catPlanta.plantaId = :plantaId " +  
 "order by txpv.txCincominutal.catMesDiaFirma.mesDiaFirmaId asc")  
 List<TxPerfilesVariable> getTxPerfilesVariableByRango(@Param("fechaInicio") Date fechaInicio, @Param("fechaFinal") Date fechaFinal,  
 @Param("plantaId") BigDecimal plantaId);  
  
  
 @Query("Select txpv from TxPerfilesVariable txpv where " +  
 "txpv.txCincominutal.catMesDiaFirma.fechaFirma >= :fechaInicio " +  
 "and txpv.txCincominutal.catMesDiaFirma.fechaFirma <= :fechaFinal " +  
 "and txpv.txCincominutal.catPlanta.plantaId = :plantaId " +  
 "and txpv.txCincominutal.catMesDiaFirma.catFirmaCincominutal.firma = :firma " +  
 "order by txpv.txCincominutal.catMesDiaFirma.mesDiaFirmaId asc")  
 List<TxPerfilesVariable> getTxPerfilesVariableByRangoAndFirmaId(@Param("fechaInicio") Date fechaInicio, @Param("fechaFinal") Date fechaFinal,  
 @Param("plantaId") BigDecimal plantaId, @Param("firma") String firma);  
  
 @Query("SELECT txPerfilesVariable FROM TxPerfilesVariable txPerfilesVariable " + "where " +  
 "txPerfilesVariable.catVariablePerfil.variablePerfilId = :variableId " +  
 "and txPerfilesVariable.catVariablePerfil.catPlanta.plantaId = :plantaId " +  
 "and txPerfilesVariable.principal = 1 " +  
 "and txPerfilesVariable.catMedidores.medidorId = :medidorId " +  
 "and txPerfilesVariable.txCincominutal.catMesDiaFirma.mesDiaFirmaId = :mesDiaFirmaId")  
 TxPerfilesVariable getTxPerfilesVariableByCatMesDiaFirma(@Param("variableId") java.math.BigDecimal variableId,  
 @Param("plantaId") BigDecimal plantaId,  
 @Param("medidorId") BigDecimal medidorId,  
 @Param("mesDiaFirmaId") BigDecimal mesDiaFirmaId) throws DyCNormaException;  
  
  
 @Query("SELECT txPerfilesVariable FROM TxPerfilesVariable txPerfilesVariable " +   
 "where " +   
 "txPerfilesVariable.catVariablePerfil.variablePerfilId = :variableId " +  
 "and txPerfilesVariable.catEstatusPerfil.estatusPerfilId = 2 " +  
 "and txPerfilesVariable.catVariablePerfil.catPlanta.plantaId = :plantaId " +  
 "and txPerfilesVariable.principal = 1 " +  
 "and txPerfilesVariable.catMedidores.medidorId = :medidorId ")  
 List<TxPerfilesVariable> getPrimeraVarValida(@Param("variableId") java.math.BigDecimal variableId,  
 @Param("plantaId") BigDecimal plantaId,  
 @Param("medidorId") BigDecimal medidorId) throws DyCNormaException;  
  
 @Query("SELECT txPerfilesVariable FROM TxPerfilesVariable txPerfilesVariable " + "where " +  
 "txPerfilesVariable.perfilId = (" +  
 "select min(txPerfilesVariableInterna.perfilId) from TxPerfilesVariable txPerfilesVariableInterna " +  
 "where " + "txPerfilesVariableInterna.catVariablePerfil.variablePerfilId = :variableId " +  
 "and txPerfilesVariableInterna.catEstatusPerfil.estatusPerfilId = 2 " +  
 "and txPerfilesVariableInterna.catVariablePerfil.catPlanta.plantaId = :plantaId " +  
 "and txPerfilesVariableInterna.principal = 1 " +  
 "and txPerfilesVariableInterna.catMedidores.medidorId = :medidorId " + ")")  
 TxPerfilesVariable getAnclaValidaCorreccion(@Param("variableId") java.math.BigDecimal variableId,  
 @Param("plantaId") BigDecimal plantaId,  
 @Param("medidorId") BigDecimal medidorId) throws DyCNormaException;  
  
 @Query("SELECT txPerfilesVariable FROM TxPerfilesVariable txPerfilesVariable " + "where " +  
 "txPerfilesVariable.perfilId = " + "(" +  
 "select max(txPerfilesVariableInterna.perfilId) from TxPerfilesVariable txPerfilesVariableInterna " +  
 "where " + "txPerfilesVariableInterna.catVariablePerfil.variablePerfilId = :variableId " +  
 "and txPerfilesVariableInterna.catEstatusPerfil.estatusPerfilId = 2 " +  
 "and txPerfilesVariableInterna.catVariablePerfil.catPlanta.plantaId = :plantaId " +  
 "and txPerfilesVariableInterna.principal = 1 " +  
 "and txPerfilesVariableInterna.catMedidores.medidorId = :medidorId)")  
 TxPerfilesVariable getPrimeraPorCincominutal(@Param("variableId") java.math.BigDecimal variableId,  
 @Param("plantaId") BigDecimal plantaId,  
 @Param("medidorId") BigDecimal medidorId) throws DyCNormaException;  
  
 @Query("select txPerfilesVariable FROM TxPerfilesVariable txPerfilesVariable where " +  
 "txPerfilesVariable.txCincominutal.cincominutalId > :cincominutalId " +  
 "and txPerfilesVariable.catVariablePerfil.variablePerfilId = :variableId " +  
 "and txPerfilesVariable.catMedidores.medidorId = :medidorId " +  
 "and txPerfilesVariable.txCincominutal.catMesDiaFirma.catMesDia.catAnioMes.anioMesId >= :anioMesId " +  
 "and txPerfilesVariable.txCincominutal.catMesDiaFirma.catMesDia.mesDiaId >= :mesDiaId " +  
 "and txPerfilesVariable.txCincominutal.catMesDiaFirma.catFirmaCincominutal.firmaId > :firmaId " +  
 "and txPerfilesVariable.txCincominutal.catPlanta.plantaId = :plantaId " +  
 "and txPerfilesVariable.principal = 1 " + "order by " +  
 "txPerfilesVariable.txCincominutal.catMesDiaFirma.mesDiaFirmaId asc")  
 List<TxPerfilesVariable> getListTxPerfilesVariable(@Param("cincominutalId") BigDecimal cincominutalId,  
 @Param("anioMesId") BigDecimal anioMesId,  
 @Param("mesDiaId") BigDecimal mesDiaId,  
 @Param("firmaId") BigDecimal firmaId,  
 @Param("variableId") BigDecimal variableId,  
 @Param("plantaId") BigDecimal plantaId,  
 @Param("medidorId") BigDecimal medidorId);  
  
 @Query("select txPerfilesVariable FROM TxPerfilesVariable txPerfilesVariable where " +  
 "txPerfilesVariable.catVariablePerfil.variablePerfilId = :variableId " +  
 "and txPerfilesVariable.catEstatusPerfil.estatusPerfilId = 1 " +  
 "and txPerfilesVariable.principal = 1 " +  
 "and txPerfilesVariable.txCincominutal.catPlanta.plantaId = :plantaId " +  
 "and txPerfilesVariable.catMedidores.medidorId = :medidorId " +   
 "order by " +  
 "txPerfilesVariable.txCincominutal.catMesDiaFirma.mesDiaFirmaId asc")  
 List<TxPerfilesVariable> getAllTxPerfilesVariable(@Param("variableId") BigDecimal variableId,  
 @Param("plantaId") BigDecimal plantaId,  
 @Param("medidorId") BigDecimal medidorId);  
   
 @Query("select txPerfilesVariable FROM TxPerfilesVariable txPerfilesVariable where " +  
 "txPerfilesVariable.txCincominutal.cincominutalId = :cincominutalId " +  
 "and txPerfilesVariable.catVariablePerfil.variablePerfilId = 7")  
 TxPerfilesVariable getPerfilForMM(@Param("cincominutalId") BigDecimal cincominutalId);  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DaoPerfilStatus.java

package mx.mbi.dgc.dao.negocio;  
  
import mx.mbi.dgc.jpa.entity.TxPerfilesVariable;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DaoPerfilStatus extends JpaRepository<TxPerfilesVariable, java.math.BigDecimal> {  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DAORangoVariable.java

package mx.mbi.dgc.dao.negocio;  
  
import java.math.BigDecimal;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatRangoVariable;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.data.repository.query.Param;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAORangoVariable extends JpaRepository<CatRangoVariable, java.math.BigDecimal> {  
 @Query("select rango FROM CatRangoVariable rango where rango.catVariable.variableId = :variableId "+  
 "and rango.catPlanta.plantaId = :plantaId ")  
 CatRangoVariable getRango(@Param("variableId") BigDecimal variableId,  
 @Param("plantaId") BigDecimal plantaId)throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DAOTags.java

package mx.mbi.dgc.dao.negocio;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatTags;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOTags extends JpaRepository<CatTags, java.math.BigDecimal> {  
 List<CatTags> findByPadreIdAndPlantaId(BigDecimal padreId, BigDecimal plantaId) throws DyCNormaException;  
 // CatTags findByIdVarStatus(BigDecimal idVarStatus) throws DyCNormaException;  
 // CatTags findByIdVarStatusAndTag(BigDecimal idVarStatus, String tag) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DaoTxUsuarioCatPlantas.java

package mx.mbi.dgc.dao.negocio;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.jpa.entity.TxUsuarioPlanta;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.data.repository.query.Param;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DaoTxUsuarioCatPlantas extends JpaRepository<TxUsuarioPlanta, java.math.BigDecimal> {  
 @Query("SELECT txUsuarioPlanta from TxUsuarioPlanta txUsuarioPlanta where " +  
 "txUsuarioPlanta.txUsuario.usuarioId = :usuarioId")  
 List<TxUsuarioPlanta> getListaPlantas(@Param("usuarioId")BigDecimal usuarioId) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DAOVariable.java

package mx.mbi.dgc.dao.negocio;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.jpa.entity.CatVariable;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOVariable extends JpaRepository<CatVariable, java.math.BigDecimal> {  
 List<CatVariable> findAllByCatPlanta(CatPlanta catPlanta) throws DyCNormaException;  
 List<CatVariable> findAllByTipoAndCatPlanta(String tipo, CatPlanta catPlanta) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\DAOVariablePerfil.java

package mx.mbi.dgc.dao.negocio;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.jpa.entity.CatVariablePerfil;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DAOVariablePerfil extends JpaRepository<CatVariablePerfil, java.math.BigDecimal> {  
 Page<CatVariablePerfil> findByVariablePerfilId(BigDecimal variablePerfilId, Pageable page) throws DyCNormaException;  
 Page<CatVariablePerfil> findByVariablePerfilContaining(String variablePerfil, Pageable page) throws DyCNormaException;  
 List<CatVariablePerfil> findAllByCatPlanta(CatPlanta catPlanta) throws DyCNormaException;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\facturacion\CustomDAOCatCompaniaPlantas.java

package mx.mbi.dgc.dao.negocio.facturacion;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.CatCompaniaPlantas;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
  
import mx.mbi.dgc.jpa.entity.CatCompania;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.jpa.entity.CatPlantaTipoFacturas;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
@Repository  
public interface CustomDAOCatCompaniaPlantas extends JpaRepository<CatCompaniaPlantas, java.math.BigDecimal> {  
 CatCompaniaPlantas findByCatPlantaAndCatCompania(CatPlanta catPlanta, CatCompania catCompania);  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\facturacion\CustomDAOCatServicioImpuestos.java

package mx.mbi.dgc.dao.negocio.facturacion;  
  
import java.util.List;  
  
import mx.mbi.dgc.jpa.entity.CatServicioImpuestos;  
import mx.mbi.dgc.jpa.entity.CatServicios;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface CustomDAOCatServicioImpuestos extends JpaRepository<CatServicioImpuestos, java.math.BigDecimal> {  
 List<CatServicioImpuestos> findByCatServicios(CatServicios catServicios) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\facturacion\CustomDAOTxDocumentos.java

package mx.mbi.dgc.dao.negocio.facturacion;  
  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxDocumentos;  
  
import org.springframework.data.domain.Page;  
import org.springframework.data.domain.Pageable;  
import org.springframework.stereotype.Repository;  
  
import java.math.BigDecimal;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.data.repository.query.Param;  
  
@Repository  
public interface CustomDAOTxDocumentos extends JpaRepository<TxDocumentos, java.math.BigDecimal> {  
  
 @Query(value =  
 "SELECT coalesce(max(to\_number(doc.folio)), 0) as folio FROM TX\_DOCUMENTOS doc where doc.serie=:serie",  
 nativeQuery = true)  
 int getMaxFolioBySerie(@Param("serie") String serie) throws Exception;  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\facturacion\DAOFacturacion.java

package mx.mbi.dgc.dao.negocio.facturacion;  
  
import java.math.BigDecimal;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;  
  
import java.text.SimpleDateFormat;  
  
import java.time.YearMonth;  
  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.Date;  
import java.util.Locale;  
  
import mx.mbi.dgc.global.Globales;  
import mx.mbi.dgc.global.Globales.Plantas;  
  
public class DAOFacturacion {  
  
 //@Value("${spring.datasource.url}")  
 private String dataSourceUrl = Globales.DATA\_SOURCE\_URL;  
 //@Value("${spring.datasource.username}")  
 private String dataSourceUser = Globales.DATA\_SOURCE\_USER;  
 //@Value("${spring.datasource.password}")  
 private String dataSourcePass = Globales.DATA\_SOURCE\_PASS;  
 //@Value("${spring.datasource.driver.class}")  
 private String getDataSourceDriver = Globales.DATA\_SOURCE\_DRIVER;  
  
 Connection conexion;  
 Statement instruccion;  
  
 private SimpleDateFormat formatoFecha = new SimpleDateFormat("dd/MM/yy", Locale.ENGLISH);  
 private SimpleDateFormat formatoFechaConHora = new SimpleDateFormat("dd/MM/yy H:mm", Locale.ENGLISH);  
  
 public BigDecimal getHorasDelMes(Integer mes) {  
 BigDecimal horasDelMes = null;  
 try {  
 instruccion = conectarABaseDeDatos();  
  
 String query = "SELECT total\_horas FROM cat\_anio\_mes cam " +  
 "WHERE cam.anio\_mes\_id = '" + mes.toString() +"'";  
  
 ResultSet resultado = instruccion.executeQuery(query);  
  
 while (resultado.next()) {  
 horasDelMes = resultado.getBigDecimal(1);  
 }  
 conexion.close();  
 } catch (Exception e){  
 e.printStackTrace();  
 }  
  
 return horasDelMes;  
 }  
  
 public BigDecimal getFDEDm(Plantas planta, YearMonth mes) {  
 BigDecimal FDEDm = null;  
 try {  
 instruccion = conectarABaseDeDatos();  
 String[] fechas = getFechasDeMesAsStrings(mes);  
  
 String query = "SELECT tvf.valor FROM TX\_VALOR\_FACTURACION tvf " +  
 "WHERE tvf.variable\_facturacion\_id = 16 " +  
 "AND tvf.fecha >= to\_date('" + fechas[0] + "', 'dd/MM/yy HH24:MI') " +  
 "AND tvf.fecha <= to\_date('" + fechas[1] + "', 'dd/MM/yy HH24:MI') " +  
 "AND tvf.planta\_id = " + planta.getID();  
 ResultSet resultado = instruccion.executeQuery(query);  
  
 if (resultado == null) {  
 String error = "No se encontro FDEDm para el mes " + mes.toString() + "\n";  
 throw new Exception(error);  
 } else {  
 while (resultado.next()){  
 FDEDm = resultado.getBigDecimal(1);  
 }  
 }  
 conexion.close();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return FDEDm;  
 }  
  
 public void setFDEDm(Plantas planta, YearMonth mes, BigDecimal FDEDm) {  
 try {  
 instruccion = conectarABaseDeDatos();  
  
 String[] fechas = getFechasDeMesAsStrings(mes);  
  
 String query = "INSERT INTO tx\_valor\_facturacion(valor\_facturacion\_id, variable\_facturacion\_id, fecha, valor, planta\_id) " +  
 "VALUES(tvf\_seq.nextval, 16, '"+ fechas[0].split(" ")[0] + "', "+FDEDm+" ," +planta.getID() +")";  
  
 instruccion.executeUpdate(query);  
 conexion.close();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 public ArrayList<BigDecimal> getCargas(Plantas planta, YearMonth mes) {  
 ArrayList<BigDecimal> cargas = null;  
 try {  
 instruccion = conectarABaseDeDatos();  
  
 String[] fechas = getFechasDeMesAsStringsSinHoras(mes);  
  
 cargas = queryDeCargasPorFechas(planta, fechas);  
  
 conexion.close();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return cargas;  
 }  
  
 public void setCFOMdm(Plantas planta, YearMonth mes, BigDecimal CFOMdm) {  
 try {  
 instruccion = conectarABaseDeDatos();  
  
 String[] fechas = getFechasDeMesAsStrings(mes);  
  
 String query = "INSERT INTO tx\_valor\_facturacion(valor\_facturacion\_id, variable\_facturacion\_id, fecha, valor, planta\_id) " +  
 "VALUES(tvf\_seq.nextval, 19, '"+ fechas[0].split(" ")[0] + "', "+ CFOMdm +" ," +planta.getID() +")";  
  
 instruccion.executeUpdate(query);  
 conexion.close();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 public ArrayList<BigDecimal> getCombustibles(Plantas planta, YearMonth mes) {  
 String[] fechas = getFechasDeMesAsStrings(mes);  
 ArrayList<BigDecimal> combustibles = new ArrayList<>();  
 try {  
 instruccion = conectarABaseDeDatos();  
  
 String query = "SELECT tc.combustible\_id FROM tx\_cincominutal tc " +  
 "LEFT JOIN cat\_mes\_dia\_firma cmdf ON tc.mes\_dia\_firma\_id = cmdf.mes\_dia\_firma\_id " +  
 "WHERE tc.planta\_id = " + planta.getID().toString() + " " +  
 "AND cmdf.fecha\_firma >= '" + fechas[0] + "' " +  
 "AND cmdf.fecha\_firma < '" + fechas[1] + "'";  
  
 ResultSet rsCombustibles = instruccion.executeQuery(query);  
  
 while (rsCombustibles.next()) {  
 combustibles.add(rsCombustibles.getBigDecimal(1));  
 }  
  
 conexion.close();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return combustibles;  
 }  
  
 public BigDecimal getArranquesDelMes(Plantas planta, YearMonth mes, int tipoDeCombustible, int tipoDeArranque) {  
 //TODO: Mock-up  
 return new BigDecimal("1");  
 }  
  
 public BigDecimal getCFOMdm(Plantas planta, YearMonth mes) {  
 String[] fechas = getFechasDeMesAsStrings(mes);  
 BigDecimal CFOMdm = null;  
 try {  
 instruccion = conectarABaseDeDatos();  
  
 String query = "SELECT tvf.valor FROM tx\_valor\_facturacion tvf " +  
 "WHERE tvf.variable\_facturacion\_id = 19 " +  
 "AND tvf.planta\_id = " + planta.getID().toString() + " " +  
 "AND tvf.fecha >= '" + fechas[0].split(" ")[0] + "' " +  
 "AND tvf.fecha <= '" + fechas[1].split(" ")[0] + "'";  
  
 ResultSet rs = instruccion.executeQuery(query);  
  
 while (rs.next()) {  
 CFOMdm = rs.getBigDecimal(1);  
 }  
  
 conexion.close();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return CFOMdm;  
 }  
  
 private ArrayList<BigDecimal> queryDeCargasPorFechas(Plantas planta, String[] fechas) throws Exception{  
  
 String query = "SELECT tc.carga\_cincominutal FROM tx\_cincominutal tc " +  
 "LEFT JOIN cat\_mes\_dia\_firma cmdf ON cmdf.mes\_dia\_firma\_id = tc.mes\_dia\_firma\_id " +  
 "WHERE tc.planta\_id = " + planta.getID() + " " +  
 "AND trunc(cmdf.FECHA\_FIRMA) BETWEEN TO\_DATE ('" + fechas[0] + "', 'dd/mm/yy') and TO\_DATE ('" + fechas[1] + "', 'dd/mm/yy')";  
 ResultSet resultados = instruccion.executeQuery(query);  
  
 return resultSetToArrayList(resultados);  
 }  
  
 String[] getFechasDeMesAsStringsSinHoras(YearMonth mes) {  
 //Obtiene el primer y último día del mes  
 Calendar[] cFechas = calcularRangoDeFechasDelMes(mes);  
  
 //Pasamos las fechas a String pasando por Date  
 return fechasToStringsSinHoras(cFechas[0], cFechas[1]);  
 }  
 String[] getFechasDeMesAsStrings(YearMonth mes) {  
 //Obtiene el primer y último día del mes  
 Calendar[] cFechas = calcularRangoDeFechasDelMes(mes);  
  
 //Pasamos las fechas a String pasando por Date  
 return fechasToStrings(cFechas[0], cFechas[1]);  
 }  
  
 private String[] fechasToStrings(Calendar fechaInicial, Calendar fechaFinal) {  
 Date dateInicial = fechaInicial.getTime();  
 Date dateFinal = fechaFinal.getTime();  
 return new String[]{formatoFecha.format(dateInicial) + " 0:00",formatoFecha.format(dateFinal) + " 23:59"};  
 }  
 private String[] fechasToStringsSinHoras(Calendar fechaInicial, Calendar fechaFinal) {  
 Date dateInicial = fechaInicial.getTime();  
 Date dateFinal = fechaFinal.getTime();  
 return new String[]{formatoFecha.format(dateInicial),formatoFecha.format(dateFinal)};  
 }  
  
 private Calendar[] calcularRangoDeFechasDelMes(YearMonth mes) {  
 Calendar[] cFechas = {Calendar.getInstance(),Calendar.getInstance()};  
 int anio = mes.getYear();  
 int nMes = mes.getMonthValue()-1;  
 cFechas[0].set(anio, nMes, 1);  
 cFechas[1].set(anio, nMes,cFechas[0].getActualMaximum(Calendar.DAY\_OF\_MONTH));  
 return cFechas;  
 }  
  
 private ArrayList<BigDecimal> resultSetToArrayList(ResultSet resultados) throws SQLException {  
 ArrayList<BigDecimal> cargas = new ArrayList<>();  
 while (resultados.next()) {  
 cargas.add(resultados.getBigDecimal(1));  
 }  
 return cargas;  
 }  
  
 BigDecimal leerVariableDeMes(Plantas planta, YearMonth mes, Integer variable) throws SQLException, ClassNotFoundException {  
 String[] fechas = getFechasDeMesAsStrings(mes);  
 instruccion = conectarABaseDeDatos();  
 BigDecimal resultadoARetornar = null;  
  
 String query = "SELECT tvf.valor FROM TX\_VALOR\_FACTURACION tvf " +  
 "WHERE tvf.variable\_facturacion\_id = "+ variable.toString() + " " +  
 "AND tvf.planta\_id = " + planta.getID().intValue() + " " +  
 "AND tvf.fecha >= to\_date('" + fechas[0] + "', 'dd/MM/yy HH24:MI') " +  
 "AND tvf.fecha <= to\_date('" + fechas[1] + "', 'dd/MM/yy HH24:MI')";  
  
 ResultSet resultados = instruccion.executeQuery(query);  
  
 while (resultados.next()) {  
 resultadoARetornar = resultados.getBigDecimal(1);  
 }  
  
 conexion.close();  
  
 return resultadoARetornar;  
 }  
  
 Statement conectarABaseDeDatos() throws ClassNotFoundException {  
 try {  
 Class.forName(getDataSourceDriver);  
  
 conexion = DriverManager.getConnection(dataSourceUrl, dataSourceUser, dataSourcePass);  
  
 return conexion.createStatement();  
 } catch (SQLException e) {  
 //e.printStackTrace();  
 System.out.println("Intentando conectar de nuevo, primer intento ...");  
 return conectarABaseDeDatos(2);  
 }  
 }  
  
 private Statement conectarABaseDeDatos(int intento) throws ClassNotFoundException{  
 final int MAX\_INTENTOS = 3;  
 try {  
 Class.forName(getDataSourceDriver);  
  
 conexion = DriverManager.getConnection(dataSourceUrl, dataSourceUser, dataSourcePass);  
  
  
 return conexion.createStatement();  
 } catch (SQLException e) {  
 if (intento <= MAX\_INTENTOS) {  
 e.printStackTrace();  
 System.out.printf("Intentando conectar de nuevo, intento %d ...", intento);  
 //YO <3 recursión  
 return conectarABaseDeDatos(++intento);  
 }  
 }  
  
 System.out.printf("No se pudo conectar despues de %d intentos\n", intento);  
 return null;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\facturacion\DAOFacturacionPCC.java

package mx.mbi.dgc.dao.negocio.facturacion;  
  
import mx.mbi.dgc.global.Globales.Plantas;  
  
import java.math.BigDecimal;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.time.YearMonth;  
import java.util.ArrayList;  
  
public class DAOFacturacionPCC extends DAOFacturacion {  
  
 public BigDecimal getCAm(Plantas planta, YearMonth mes) {  
 //TODO: maqueteado, falta implementación real  
 BigDecimal CAm = null;  
 try {  
 CAm = leerVariableDeMes(planta, mes, 17);  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return CAm;  
 }  
  
 public BigDecimal getPPGm(Plantas planta, YearMonth mes) {  
 //TODO: maqueteado, falta implementación real  
 BigDecimal PPGm = null;  
 try {  
 PPGm = leerVariableDeMes(planta, mes, 18);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return PPGm;  
 }  
  
 public ArrayList<BigDecimal> getPCIm(Plantas planta, YearMonth mes) {  
  
 return leerCincominutalVarDelMes(planta, mes, 6, 15);  
 }  
  
 public ArrayList<BigDecimal> getPCSm(Plantas planta, YearMonth mes) {  
  
 return leerCincominutalVarDelMes(planta, mes, 7, 16);  
 }  
  
  
 public ArrayList<BigDecimal> getEnergias(Plantas planta, YearMonth mes) {  
 String[] fechas = getFechasDeMesAsStrings(mes);  
 ArrayList<BigDecimal> energias = new ArrayList<>();  
 try {  
 instruccion = conectarABaseDeDatos();  
  
 String query = "SELECT tc.energia\_activa FROM tx\_cincominutal tc " +  
 "LEFT JOIN cat\_mes\_dia\_firma cmdf ON cmdf.mes\_dia\_firma\_id = tc.mes\_dia\_firma\_id " +  
 "WHERE tc.planta\_id = " + planta.getID().toString() + " " +  
 "AND cmdf.fecha\_firma >= '" + fechas[0] + "' " +  
 "AND cmdf.fecha\_firma < '" + fechas[1] + "'";  
  
 System.out.printf("QUERY: %S\n", query);  
 ResultSet rsEnergias = instruccion.executeQuery(query);  
  
 while (rsEnergias.next()) {  
 energias.add(rsEnergias.getBigDecimal(1));  
 }  
  
 conexion.close();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return energias;  
 }  
  
 public ArrayList<BigDecimal> leerCincominutalVarDelMes(Plantas planta, Integer combustible, YearMonth mes, Integer... variables) {  
 String[] fechas = getFechasDeMesAsStrings(mes);  
 ArrayList<BigDecimal> valores = null;  
  
 try {  
 instruccion = conectarABaseDeDatos();  
  
 StringBuilder query = new StringBuilder("SELECT tcv.valor\_norma FROM tx\_cincominutal\_var tcv " +  
 "LEFT JOIN cat\_variable cv ON cv.variable\_id = tcv.variable\_id " +  
 "LEFT JOIN tx\_cincominutal tc ON tc.cincominutal\_id = tcv.cincominutal\_id " +  
 "LEFT JOIN cat\_mes\_dia\_firma cmdf ON cmdf.mes\_dia\_firma\_id = tc.mes\_dia\_firma\_id " +  
 "WHERE cv.planta\_id = " + planta.getID().toString() + " " +  
 "AND tc.combustible\_id = " + combustible.toString() + " " +  
 "AND cmdf.fecha\_firma >= '" + fechas[0] + "' " +  
 "AND cmdf.fecha\_firma <= '" + fechas[1] + "' " +  
 "AND (");  
  
 agregarListaDeVariablesAlQuery(query, variables);  
 System.out.printf("QUERY: %s\n", query);  
 ResultSet resultados = instruccion.executeQuery(query.toString());  
 valores = resultSetToArrayListDeValores(resultados);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return valores;  
 }  
  
 private ArrayList<BigDecimal> leerCincominutalVarDelMes(Plantas planta, YearMonth mes, Integer... variables) {  
 String[] fechas = getFechasDeMesAsStrings(mes);  
 ArrayList<BigDecimal> valores = null;  
  
 try {  
 instruccion = conectarABaseDeDatos();  
  
 StringBuilder query = new StringBuilder("SELECT tcv.valor\_norma FROM tx\_cincominutal\_var tcv " +  
 "LEFT JOIN cat\_variable cv ON cv.variable\_id = tcv.variable\_id " +  
 "LEFT JOIN tx\_cincominutal tc ON tc.cincominutal\_id = tcv.cincominutal\_id " +  
 "LEFT JOIN cat\_mes\_dia\_firma cmdf ON cmdf.mes\_dia\_firma\_id = tc.mes\_dia\_firma\_id " +  
 "WHERE cv.planta\_id = " + planta.getID().toString() + " " +  
 "AND cmdf.fecha\_firma >= '" + fechas[0] + "' " +  
 "AND cmdf.fecha\_firma <= '" + fechas[1] + "' " +  
 "AND (");  
  
 agregarListaDeVariablesAlQuery(query, variables);  
 //System.out.printf("QUERY: %s\n", query);  
 ResultSet resultados = instruccion.executeQuery(query.toString());  
 valores = resultSetToArrayListDeValores(resultados);  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return valores;  
 }  
  
 private void agregarListaDeVariablesAlQuery(StringBuilder query, Integer[] variables) {  
 int iteracion = 0;  
 for (Integer variable :  
 variables) {  
 iteracion++;  
 query.append("cv.variable\_id = ").append(variable.toString()).append(" ");  
 if (!(iteracion >= variables.length)) {  
 query.append("OR ");  
 }  
 }  
  
 query.append(") ");  
 }  
  
 private ArrayList<BigDecimal> resultSetToArrayListDeValores(ResultSet set) throws SQLException {  
 ArrayList<BigDecimal> valores = new ArrayList<>();  
  
 while (set.next()) {  
 valores.add(set.getBigDecimal(1));  
 }  
  
 return valores;  
 }  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\facturacion\DAONegocioCatAnioMes.java

package mx.mbi.dgc.dao.negocio.facturacion;  
  
import mx.mbi.dgc.jpa.entity.CatAnioMes;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.data.jpa.repository.Query;  
import org.springframework.data.repository.query.Param;  
  
import java.math.BigDecimal;  
  
  
public interface DAONegocioCatAnioMes extends JpaRepository<CatAnioMes, BigDecimal> {  
  
 @Query("SELECT cam.totalHoras FROM CatAnioMes cam " +  
 "WHERE cam.anioMesId = :numeroDeMes")  
 BigDecimal getHorasDelMes(@Param("numeroDeMes") Integer numeroDeMes); //El número de mes desde enero de 2014 (enero 2014 vale 1)  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\facturacion\DaoNegocioCatComponentes.java

package mx.mbi.dgc.dao.negocio.facturacion;  
  
import java.util.List;  
  
import mx.mbi.dgc.jpa.entity.CatComponentes;  
import mx.mbi.dgc.jpa.entity.CatFormulaComponentes;  
import mx.mbi.dgc.jpa.entity.TxFormulas;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DaoNegocioCatComponentes extends JpaRepository<CatComponentes, java.math.BigDecimal> {  
 public CatComponentes findByComponente(String componente);  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\facturacion\DaoNegocioCatFormulaComponentes.java

package mx.mbi.dgc.dao.negocio.facturacion;  
  
import java.util.List;  
  
import mx.mbi.dgc.jpa.entity.CatFormulaComponentes;  
import mx.mbi.dgc.jpa.entity.TxFormulas;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DaoNegocioCatFormulaComponentes extends JpaRepository<CatFormulaComponentes, java.math.BigDecimal> {  
 public List<CatFormulaComponentes> findByTxFormulas(TxFormulas txFormulas);  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\facturacion\DaoNegocioTxFormulas.java

package mx.mbi.dgc.dao.negocio.facturacion;  
  
import mx.mbi.dgc.jpa.entity.CatAnioMes;  
import mx.mbi.dgc.jpa.entity.TxEscenario;  
import mx.mbi.dgc.jpa.entity.TxFormulas;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DaoNegocioTxFormulas extends JpaRepository<TxFormulas, java.math.BigDecimal> {  
 public TxFormulas findByCatAnioMesAndTxEscenarioAndFormula(CatAnioMes catAnioMes, TxEscenario txEscenario, String formula);  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\tags\DaoNegocioTags.java

package mx.mbi.dgc.dao.negocio.tags;  
  
import mx.mbi.dgc.jpa.entity.CatTags;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DaoNegocioTags extends JpaRepository<CatTags, java.math.BigDecimal>/\*, TagsInterface\*/ {  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\tags\package-info.java

package mx.mbi.dgc.dao.negocio.tags;

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\tags\TagsImpl.java

package mx.mbi.dgc.dao.negocio.tags;  
  
import java.math.BigDecimal;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import javax.persistence.EntityManager;  
import javax.persistence.PersistenceContext;  
import javax.persistence.Query;  
  
import javax.script.ScriptEngine;  
import javax.script.ScriptEngineManager;  
  
import javax.transaction.Transactional;  
  
import mx.mbi.dgc.dominio.negocio.dyc.HashTags;  
import mx.mbi.dgc.dominio.negocio.dyc.TagValue;  
import mx.mbi.dgc.jpa.entity.CatTags;  
  
import org.hibernate.transform.ResultTransformer;  
  
import org.springframework.stereotype.Component;  
  
@Transactional  
@Component  
public class TagsImpl implements TagsInterface {  
  
 @PersistenceContext  
 private EntityManager em;  
  
 /\*@SqlResultSetMapping(  
 name = "AuthorValueMapping",  
 classes = @ConstructorResult(  
 targetClass = HashTags.class,  
 columns = {  
 @ColumnResult(name = "id", type = Long.class),  
 @ColumnResult(name = "firstname"),  
 @ColumnResult(name = "lastname"),jav  
 @ColumnResult(name = "numBooks", type = Long.class)}))\*/  
  
  
 @Override  
 public List<TagValue> ejecutaQuery(CatTags catTag, String fechaInicio, String fechaFinal) {  
  
 ScriptEngineManager manager = new ScriptEngineManager();  
 ScriptEngine engine = manager.getEngineByName("js");  
  
 List<HashTags> retorno = new ArrayList<>();  
 String query = catTag.getQueryInformacion();  
   
 //query = query.replace(":fechaInicio", fechaInicio);  
 //query = query.replace(":fechaFin", fechaFinal);  
  
 // System.out.println(catTag.getTag());  
   
 //em.createNativeQuery("alter session set nls\_date\_format = 'DD/MM/YYYY'").executeUpdate();  
   
 System.out.println("fecha inicio:"+fechaInicio);  
 System.out.println("fecha fin:"+fechaFinal);  
 System.out.println("query:"+query);  
 System.out.println(catTag.getTagId());  
 Query q = em.createNativeQuery(query);  
 q.setParameter("fechaInicio", fechaInicio);  
 q.setParameter("fechaFin", fechaFinal);  
   
   
 //List<HashTags> listHashTags =  
 List<TagValue> tagValues = q.unwrap(org.hibernate.query.Query.class)  
 .setResultTransformer(new ResultTransformer() {  
 //return Collections.emptyList();  
 @Override  
 public Object transformTuple(Object[] object, String[] string) {  
 try {  
 TagValue tagValue = new TagValue();  
 tagValue.setTagKey((String) object[0] + " " + (String) object[1]);  
 List<String> values = new ArrayList<>();  
 List<String> colors = new ArrayList<>();  
 String valueGraph = "";  
 int index = 0;  
 for (Object value : object) {  
 if (index >= 3) {  
 if (string[index].toUpperCase().contains("VALUE")) {  
 if (value != null) {  
  
 // System.out.println("no nulo " + value + " y clase:" + value.getClass());  
 if (value.getClass() == BigDecimal.class) {  
 // System.out.println("bigdecimal");  
 if (index == 3) {  
 engine.put("X", value);  
 // System.out.println("valor: " + value);  
 // System.out.println("expresión: " + (String) object[2]);  
 if (object[2] != null) {  
 Object valorEscalado = engine.eval((String) object[2]);  
 valueGraph = String.valueOf((Double) valorEscalado);  
 } else {  
 valueGraph = String.valueOf(value);  
 }  
 // System.out.println("valor escalado: " + valorEscalado);  
 }  
 BigDecimal val = (BigDecimal) value;  
 values.add(val.toString());  
 }  
 if (value.getClass() == String.class) {  
 if (index == 3) {  
 engine.put("X", value);  
 // System.out.println("valor: " + value);  
 // System.out.println("expresión: " + (String) object[2]);  
  
 if (object[2] != null) {  
 Object valorEscalado = engine.eval((String) object[2]);  
 valueGraph = String.valueOf((Double) valorEscalado);  
 } else {  
 valueGraph = String.valueOf(value);  
 }  
 // System.out.println("valor escalado: " + valorEscalado);  
 try {  
 BigDecimal val = new BigDecimal((String) value);  
 values.add(val.toString());  
 } catch (Exception e) {  
 // System.out.println("falló conversión");  
 values.add((String) value);  
 }  
 } else {  
 String val = (String) value;  
 values.add(val);  
 }  
 }  
 } else  
 values.add(null);  
 } else {  
 if (string[index].toUpperCase().contains("COLOR")) {  
 colors.add(String.valueOf(value));  
 }  
 }  
 }  
 index++;  
 }  
 tagValue.setTagValues(values);  
 tagValue.setNormalizedValue(valueGraph);  
 tagValue.setColors(colors);  
  
 return tagValue;  
 } catch (Exception e) {  
 e.printStackTrace();  
 return null;  
 }  
 }  
  
 @Override  
 public List transformList(List list) {  
 // System.out.println("Entré a regresa list");  
 return list;  
 }  
 })  
 .getResultList();  
  
  
 return tagValues;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\dao\negocio\tags\TagsInterface.java

package mx.mbi.dgc.dao.negocio.tags;  
  
import java.util.List;  
  
import mx.mbi.dgc.dominio.negocio.dyc.TagValue;  
import mx.mbi.dgc.jpa.entity.CatTags;  
  
public interface TagsInterface {  
 public List<TagValue> ejecutaQuery(CatTags catTag, String fechaInicio, String FechaFinal);  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\excepcion\CargaException.java

package mx.mbi.dgc.excepcion;  
  
public class CargaException extends Exception {  
 @SuppressWarnings("compatibility:-8417475342587208891")  
 private static final long serialVersionUID = -1127219831870567637L;  
  
 public CargaException(String message, Throwable cause) {  
 super(message, cause);  
 }  
   
 public CargaException(String message) {  
 super(message);  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\excepcion\CustomAsyncExceptionHandler.java

package mx.mbi.dgc.excepcion;  
  
import java.lang.reflect.Method;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.aop.interceptor.AsyncUncaughtExceptionHandler;  
  
public class CustomAsyncExceptionHandler implements AsyncUncaughtExceptionHandler {  
 private static final Logger LOG = LogManager.getLogger(CustomAsyncExceptionHandler.class);  
   
 @Override  
 public void handleUncaughtException(Throwable throwable, Method method, Object... obj) {  
 LOG.error("CustomAsyncExceptionHandler.Exception message - " + throwable.getMessage());  
 LOG.error("CustomAsyncExceptionHandler.Method name - " + method.getName());  
 for (Object param : obj) {  
 LOG.error("CustomAsyncExceptionHandler.Parameter value - " + param);  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\excepcion\DyCNormaException.java

package mx.mbi.dgc.excepcion;  
  
public class DyCNormaException extends Exception {  
 @SuppressWarnings("compatibility:7656030030558397185")  
 private static final long serialVersionUID = 8526710103339034324L;  
  
 public DyCNormaException(String message, Throwable cause) {  
 super(message, cause);  
 }  
   
 public DyCNormaException(String message) {  
 super(message);  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\global\Globales.java

package mx.mbi.dgc.global;  
  
  
import java.math.BigDecimal;  
  
/\*\* Contiene las definiciones de constantes globales para safe en MSA \*/  
public class Globales {  
   
 //Esta clase no se instancia  
 private Globales(){  
 }  
   
   
 //CONSTANTES DE PATH  
 //public static final String PATH\_EXCEL = "C:\\Programas\\nginx-1.14.0\\html\\dgc\\excel\\";  
 //public static final String PATH\_EXCEL = "/Programas/qa/excel";  
 //public static final String PATH\_EXCEL = "/etc/nginx/html/dgc/excel/";  
 //public static final String PATH\_MM\_EST = "/Pruebas/FTP/out/EST/mm/";  
 //public static final String PATH\_MM\_EAT = "/Pruebas/FTP/out/EAT/mm/";  
 public static final String PATH\_MM\_EST = "/Programas/dev/Pruebas/EST/mm/";  
 public static final String PATH\_MM\_EAT = "/Programas/dev/Pruebas/EAT/mm/";  
 public static final String PATH\_SUPPORT\_FILE = "/Pruebas/Bitacora/Soporte/";  
 //public static final String PATH\_DYC\_EX = "/Pruebas/FTP/out/DYC/";  
 //public static final String PATH\_DYC\_EX = "/Programas/qa/excel";  
 public static final String PATH\_DYC\_EX = "/etc/nginx/html/dgc/excel/";  
 //public static final String PATH\_DYC\_EX = "C:\\Programas\\nginx-1.14.0\\html\\dgc\\excel";  
  
// public static final String PATH\_FACTURACION\_PLANTILLAS = "/Programas/dev/facturas/plantillas/";  
// public static final String PATH\_FACTURACION\_DOCUMENTOS = "/Programas/dev/facturas/documentos/";  
// public static final String PATH\_FACTURACION\_ZIPS = "/Programas/dev/facturas/zips/";  
 public static final String PATH\_FACTURACION\_PLANTILLAS = "/Programas/dev/facturas/plantillas/";  
 public static final String PATH\_FACTURACION\_DOCUMENTOS = "/Programas/dev/facturas/documentos/";  
 public static final String PATH\_FACTURACION\_ZIPS = "/Programas/dev/facturas/zips/";  
  
   
 //CONSTANTES DE CORRESPONDENCIA INT-NOMBRE  
 public static final int EAT = 1;//1 = Aguila  
 public static final int EST = 2;//2 = Sol  
   
 //CONSTANTES DE CAPACIDAD DE ENERGIA  
 public static final int GAS\_CAPACITY=495;  
 public static final int OIL\_CAPACITY=405;  
  
 //CONSTANTES DE TIMBRADO  
 public static final String RUTA\_CERTIFICADO\_LINUX = "/Programas/dev/Pruebas/certificados/";  
 public static final String RUTA\_CERTIFICADO\_WINDOWS = "C:\\Programas\\Certificados\\";  
  
 //Listado de plantas y su valor correspondiente en BigDecimal  
  
 public static enum Plantas {  
 Invalida(new BigDecimal("-1")),  
 Aguila(new BigDecimal("1")),  
 Sol(new BigDecimal("2"));  
  
 private final BigDecimal ID;  
  
 Plantas(BigDecimal ID){  
 this.ID = ID;  
 }  
  
 public BigDecimal getID() {  
 return ID;  
 }  
 }  
  
 public static Plantas getPlanta(int plantaID){  
 for (Plantas planta: Plantas.values()) {  
 if (planta.getID().equals(new BigDecimal(plantaID))){  
 return planta;  
 }  
 }  
 return Plantas.Invalida;  
 }  
  
 //Constantes para JDBC  
 public static String DATA\_SOURCE\_URL = "jdbc:oracle:thin:@200.52.85.139:1521/devicollab.dgc.com";  
 public static String DATA\_SOURCE\_USER = "QADGC";  
 public static String DATA\_SOURCE\_PASS = "rxVLet4P";  
 public static String DATA\_SOURCE\_DRIVER = "oracle.jdbc.driver.OracleDriver";  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatAnioMes.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import java.util.HashSet;  
import java.util.Set;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatAnioMes generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_ANIO\_MES"  
)  
public class CatAnioMes implements java.io.Serializable {  
  
  
 private BigDecimal anioMesId;  
 private CatPlanta catPlanta;  
 private String anioMes;  
 private BigDecimal totalDias;  
 private BigDecimal totalHoras;  
 private Set<CatMesDia> catMesDias = new HashSet<CatMesDia>(0);  
 private Set<TxFormulas> txFormulases = new HashSet<TxFormulas>(0);  
  
 public CatAnioMes() {  
 }  
  
   
 public CatAnioMes(CatPlanta catPlanta, String anioMes, BigDecimal totalDias, BigDecimal totalHoras) {  
 this.catPlanta = catPlanta;  
 this.anioMes = anioMes;  
 this.totalDias = totalDias;  
 this.totalHoras = totalHoras;  
 }  
 public CatAnioMes(CatPlanta catPlanta, String anioMes, BigDecimal totalDias, BigDecimal totalHoras, Set<CatMesDia> catMesDias, Set<TxFormulas> txFormulases) {  
 this.catPlanta = catPlanta;  
 this.anioMes = anioMes;  
 this.totalDias = totalDias;  
 this.totalHoras = totalHoras;  
 this.catMesDias = catMesDias;  
 this.txFormulases = txFormulases;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="ANIO\_MES\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getAnioMesId() {  
 return this.anioMesId;  
 }  
   
 public void setAnioMesId(BigDecimal anioMesId) {  
 this.anioMesId = anioMesId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="PLANTA\_ID", nullable=false)  
 public CatPlanta getCatPlanta() {  
 return this.catPlanta;  
 }  
   
 public void setCatPlanta(CatPlanta catPlanta) {  
 this.catPlanta = catPlanta;  
 }  
  
   
 @Column(name="ANIO\_MES", nullable=false, length=100)  
 public String getAnioMes() {  
 return this.anioMes;  
 }  
   
 public void setAnioMes(String anioMes) {  
 this.anioMes = anioMes;  
 }  
  
   
 @Column(name="TOTAL\_DIAS", nullable=false, precision=20, scale=6)  
 public BigDecimal getTotalDias() {  
 return this.totalDias;  
 }  
   
 public void setTotalDias(BigDecimal totalDias) {  
 this.totalDias = totalDias;  
 }  
  
   
 @Column(name="TOTAL\_HORAS", nullable=false, precision=20, scale=6)  
 public BigDecimal getTotalHoras() {  
 return this.totalHoras;  
 }  
   
 public void setTotalHoras(BigDecimal totalHoras) {  
 this.totalHoras = totalHoras;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catAnioMes")  
 public Set<CatMesDia> getCatMesDias() {  
 return this.catMesDias;  
 }  
   
 public void setCatMesDias(Set<CatMesDia> catMesDias) {  
 this.catMesDias = catMesDias;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catAnioMes")  
 public Set<TxFormulas> getTxFormulases() {  
 return this.txFormulases;  
 }  
   
 public void setTxFormulases(Set<TxFormulas> txFormulases) {  
 this.txFormulases = txFormulases;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatCapacidad.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatCapacidad generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_CAPACIDAD"  
)  
public class CatCapacidad implements java.io.Serializable {  
  
  
 private BigDecimal capacidadId;  
 private BigDecimal capacidad;  
 private BigDecimal rangoMinimo;  
 private BigDecimal rangoMaximo;  
 private Set<CatCurvasCorreccion> catCurvasCorreccions = new HashSet<CatCurvasCorreccion>(0);  
 private Set<CatLimitesDominio> catLimitesDominios = new HashSet<CatLimitesDominio>(0);  
  
 public CatCapacidad() {  
 }  
  
   
 public CatCapacidad(BigDecimal capacidad, BigDecimal rangoMinimo, BigDecimal rangoMaximo) {  
 this.capacidad = capacidad;  
 this.rangoMinimo = rangoMinimo;  
 this.rangoMaximo = rangoMaximo;  
 }  
 public CatCapacidad(BigDecimal capacidad, BigDecimal rangoMinimo, BigDecimal rangoMaximo, Set<CatCurvasCorreccion> catCurvasCorreccions, Set<CatLimitesDominio> catLimitesDominios) {  
 this.capacidad = capacidad;  
 this.rangoMinimo = rangoMinimo;  
 this.rangoMaximo = rangoMaximo;  
 this.catCurvasCorreccions = catCurvasCorreccions;  
 this.catLimitesDominios = catLimitesDominios;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="CAPACIDAD\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getCapacidadId() {  
 return this.capacidadId;  
 }  
   
 public void setCapacidadId(BigDecimal capacidadId) {  
 this.capacidadId = capacidadId;  
 }  
  
   
 @Column(name="CAPACIDAD", nullable=false, precision=22, scale=0)  
 public BigDecimal getCapacidad() {  
 return this.capacidad;  
 }  
   
 public void setCapacidad(BigDecimal capacidad) {  
 this.capacidad = capacidad;  
 }  
  
   
 @Column(name="RANGO\_MINIMO", nullable=false, precision=22, scale=0)  
 public BigDecimal getRangoMinimo() {  
 return this.rangoMinimo;  
 }  
   
 public void setRangoMinimo(BigDecimal rangoMinimo) {  
 this.rangoMinimo = rangoMinimo;  
 }  
  
   
 @Column(name="RANGO\_MAXIMO", nullable=false, precision=22, scale=0)  
 public BigDecimal getRangoMaximo() {  
 return this.rangoMaximo;  
 }  
   
 public void setRangoMaximo(BigDecimal rangoMaximo) {  
 this.rangoMaximo = rangoMaximo;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catCapacidad")  
 public Set<CatCurvasCorreccion> getCatCurvasCorreccions() {  
 return this.catCurvasCorreccions;  
 }  
   
 public void setCatCurvasCorreccions(Set<CatCurvasCorreccion> catCurvasCorreccions) {  
 this.catCurvasCorreccions = catCurvasCorreccions;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catCapacidad")  
 public Set<CatLimitesDominio> getCatLimitesDominios() {  
 return this.catLimitesDominios;  
 }  
   
 public void setCatLimitesDominios(Set<CatLimitesDominio> catLimitesDominios) {  
 this.catLimitesDominios = catLimitesDominios;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatCargoFijo.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatCargoFijo generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_CARGO\_FIJO"  
)  
public class CatCargoFijo implements java.io.Serializable {  
  
  
 private BigDecimal cargoFijoId;  
 private CatSubCargos catSubCargos;  
 private CatMesCargo catMesCargo;  
 private BigDecimal valor;  
 private boolean principal;  
 private String pathSoporte;  
 private String nombreEvidencia;  
  
 public CatCargoFijo() {  
 }  
  
   
 public CatCargoFijo(CatSubCargos catSubCargos, CatMesCargo catMesCargo, BigDecimal valor, boolean principal) {  
 this.catSubCargos = catSubCargos;  
 this.catMesCargo = catMesCargo;  
 this.valor = valor;  
 this.principal = principal;  
 }  
 public CatCargoFijo(CatSubCargos catSubCargos, CatMesCargo catMesCargo, BigDecimal valor, boolean principal, String pathSoporte, String nombreEvidencia) {  
 this.catSubCargos = catSubCargos;  
 this.catMesCargo = catMesCargo;  
 this.valor = valor;  
 this.principal = principal;  
 this.pathSoporte = pathSoporte;  
 this.nombreEvidencia = nombreEvidencia;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="CARGO\_FIJO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getCargoFijoId() {  
 return this.cargoFijoId;  
 }  
   
 public void setCargoFijoId(BigDecimal cargoFijoId) {  
 this.cargoFijoId = cargoFijoId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="SUB\_CARGO\_ID", nullable=false)  
 public CatSubCargos getCatSubCargos() {  
 return this.catSubCargos;  
 }  
   
 public void setCatSubCargos(CatSubCargos catSubCargos) {  
 this.catSubCargos = catSubCargos;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="MES\_ID", nullable=false)  
 public CatMesCargo getCatMesCargo() {  
 return this.catMesCargo;  
 }  
   
 public void setCatMesCargo(CatMesCargo catMesCargo) {  
 this.catMesCargo = catMesCargo;  
 }  
  
   
 @Column(name="VALOR", nullable=false, precision=22, scale=0)  
 public BigDecimal getValor() {  
 return this.valor;  
 }  
   
 public void setValor(BigDecimal valor) {  
 this.valor = valor;  
 }  
  
   
 @Column(name="PRINCIPAL", nullable=false, precision=1, scale=0)  
 public boolean isPrincipal() {  
 return this.principal;  
 }  
   
 public void setPrincipal(boolean principal) {  
 this.principal = principal;  
 }  
  
   
 @Column(name="PATH\_SOPORTE", length=250)  
 public String getPathSoporte() {  
 return this.pathSoporte;  
 }  
   
 public void setPathSoporte(String pathSoporte) {  
 this.pathSoporte = pathSoporte;  
 }  
  
   
 @Column(name="NOMBRE\_EVIDENCIA", length=100)  
 public String getNombreEvidencia() {  
 return this.nombreEvidencia;  
 }  
   
 public void setNombreEvidencia(String nombreEvidencia) {  
 this.nombreEvidencia = nombreEvidencia;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatCargos.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatCargos generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_CARGOS"  
)  
public class CatCargos implements java.io.Serializable {  
  
  
 private BigDecimal cargoId;  
 private CatPlanta catPlanta;  
 private String cargo;  
 private Set<CatSubCargos> catSubCargoses = new HashSet<CatSubCargos>(0);  
  
 public CatCargos() {  
 }  
  
   
 public CatCargos(String cargo) {  
 this.cargo = cargo;  
 }  
 public CatCargos(CatPlanta catPlanta, String cargo, Set<CatSubCargos> catSubCargoses) {  
 this.catPlanta = catPlanta;  
 this.cargo = cargo;  
 this.catSubCargoses = catSubCargoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="CARGO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getCargoId() {  
 return this.cargoId;  
 }  
   
 public void setCargoId(BigDecimal cargoId) {  
 this.cargoId = cargoId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="PLANTA\_ID")  
 public CatPlanta getCatPlanta() {  
 return this.catPlanta;  
 }  
   
 public void setCatPlanta(CatPlanta catPlanta) {  
 this.catPlanta = catPlanta;  
 }  
  
   
 @Column(name="CARGO", nullable=false, length=250)  
 public String getCargo() {  
 return this.cargo;  
 }  
   
 public void setCargo(String cargo) {  
 this.cargo = cargo;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catCargos")  
 public Set<CatSubCargos> getCatSubCargoses() {  
 return this.catSubCargoses;  
 }  
   
 public void setCatSubCargoses(Set<CatSubCargos> catSubCargoses) {  
 this.catSubCargoses = catSubCargoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatClasificacion.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatClasificacion generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_CLASIFICACION"  
)  
public class CatClasificacion implements java.io.Serializable {  
  
  
 private BigDecimal clasificacionId;  
 private String color;  
 private BigDecimal fadd;  
 //private Set<TxEvento> txEventos = new HashSet<TxEvento>(0);  
  
 public CatClasificacion() {  
 }  
  
   
 public CatClasificacion(String color, BigDecimal fadd) {  
 this.color = color;  
 this.fadd = fadd;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="CLASIFICACION\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getClasificacionId() {  
 return this.clasificacionId;  
 }  
   
 public void setClasificacionId(BigDecimal clasificacionId) {  
 this.clasificacionId = clasificacionId;  
 }  
  
   
 @Column(name="COLOR", nullable=false, length=100)  
 public String getColor() {  
 return this.color;  
 }  
   
 public void setColor(String color) {  
 this.color = color;  
 }  
  
   
 @Column(name="FADD", nullable=false, precision=22, scale=0)  
 public BigDecimal getFadd() {  
 return this.fadd;  
 }  
   
 public void setFadd(BigDecimal fadd) {  
 this.fadd = fadd;  
 }  
  
  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatClasificacionEvento.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
  
import java.math.BigDecimal;  
  
import java.util.Set;  
  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlTransient;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_CLASIFICACION\_EVENTOS")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatClasificacionEvento.findAll", query = "SELECT c FROM CatClasificacionEvento c")  
 , @NamedQuery(name = "CatClasificacionEvento.findByClasificacionEventoId", query = "SELECT c FROM CatClasificacionEvento c WHERE c.clasificacionEventoId = :clasificacionEventoId")  
 , @NamedQuery(name = "CatClasificacionEvento.findByClasificacionEvento", query = "SELECT c FROM CatClasificacionEvento c WHERE c.clasificacionEvento = :clasificacionEvento")})  
public class CatClasificacionEvento implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "CLASIFICACION\_EVENTO\_ID")  
 private BigDecimal clasificacionEventoId;  
 @Size(max = 150)  
 @Column(name = "CLASIFICACION\_EVENTO")  
 private String clasificacionEvento;  
 @OneToMany(mappedBy = "clasificacionEventoId")  
 private Set<CatConfiguracionEventos> catConfiguracionEventosSet;  
  
 public CatClasificacionEvento() {  
 }  
  
 public CatClasificacionEvento(BigDecimal clasificacionEventoId) {  
 this.clasificacionEventoId = clasificacionEventoId;  
 }  
  
 public BigDecimal getClasificacionEventoId() {  
 return clasificacionEventoId;  
 }  
  
 public void setClasificacionEventoId(BigDecimal clasificacionEventoId) {  
 this.clasificacionEventoId = clasificacionEventoId;  
 }  
  
 public String getClasificacionEvento() {  
 return clasificacionEvento;  
 }  
  
 public void setClasificacionEvento(String clasificacionEvento) {  
 this.clasificacionEvento = clasificacionEvento;  
 }  
  
 @XmlTransient  
 public Set<CatConfiguracionEventos> getCatConfiguracionEventosSet() {  
 return catConfiguracionEventosSet;  
 }  
  
 public void setCatConfiguracionEventosSet(Set<CatConfiguracionEventos> catConfiguracionEventosSet) {  
 this.catConfiguracionEventosSet = catConfiguracionEventosSet;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (clasificacionEventoId != null ? clasificacionEventoId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatClasificacionEvento)) {  
 return false;  
 }  
 CatClasificacionEvento other = (CatClasificacionEvento) object;  
 if ((this.clasificacionEventoId == null && other.clasificacionEventoId != null) || (this.clasificacionEventoId != null && !this.clasificacionEventoId.equals(other.clasificacionEventoId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatClasificacionEvento[ clasificacionEventoId=" + clasificacionEventoId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatClasificacionEventos.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import java.math.BigInteger;  
import java.util.Date;  
import java.util.Set;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlTransient;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_CLASIFICACION\_EVENTOS")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatClasificacionEventos.findAll", query = "SELECT c FROM CatClasificacionEventos c")  
 , @NamedQuery(name = "CatClasificacionEventos.findByClasificacionEventoId", query = "SELECT c FROM CatClasificacionEventos c WHERE c.clasificacionEventoId = :clasificacionEventoId")  
 , @NamedQuery(name = "CatClasificacionEventos.findByNombre", query = "SELECT c FROM CatClasificacionEventos c WHERE c.nombre = :nombre")  
 , @NamedQuery(name = "CatClasificacionEventos.findBySiglas", query = "SELECT c FROM CatClasificacionEventos c WHERE c.siglas = :siglas")  
 , @NamedQuery(name = "CatClasificacionEventos.findByDescripcion", query = "SELECT c FROM CatClasificacionEventos c WHERE c.descripcion = :descripcion")  
 , @NamedQuery(name = "CatClasificacionEventos.findByRequiereOt", query = "SELECT c FROM CatClasificacionEventos c WHERE c.requiereOt = :requiereOt")  
 , @NamedQuery(name = "CatClasificacionEventos.findByEsFalla", query = "SELECT c FROM CatClasificacionEventos c WHERE c.esFalla = :esFalla")  
 , @NamedQuery(name = "CatClasificacionEventos.findByActivo", query = "SELECT c FROM CatClasificacionEventos c WHERE c.activo = :activo")  
 , @NamedQuery(name = "CatClasificacionEventos.findByFechaRegistro", query = "SELECT c FROM CatClasificacionEventos c WHERE c.fechaRegistro = :fechaRegistro")  
 , @NamedQuery(name = "CatClasificacionEventos.findByUsuarioRegistro", query = "SELECT c FROM CatClasificacionEventos c WHERE c.usuarioRegistro = :usuarioRegistro")  
 , @NamedQuery(name = "CatClasificacionEventos.findByFechaModificacion", query = "SELECT c FROM CatClasificacionEventos c WHERE c.fechaModificacion = :fechaModificacion")  
 , @NamedQuery(name = "CatClasificacionEventos.findByUsuarioModificoId", query = "SELECT c FROM CatClasificacionEventos c WHERE c.usuarioModificoId = :usuarioModificoId")})  
public class CatClasificacionEventos implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "CLASIFICACION\_EVENTO\_ID")  
 private BigDecimal clasificacionEventoId;  
 @Size(max = 1020)  
 @Column(name = "NOMBRE")  
 private String nombre;  
 @Size(max = 4000)  
 @Column(name = "SIGLAS")  
 private String siglas;  
 @Size(max = 4000)  
 @Column(name = "DESCRIPCION")  
 private String descripcion;  
 @Column(name = "\u00bfREQUIERE\_OT")  
 private BigInteger requiereOt;  
 @Column(name = "\u00bfES\_FALLA")  
 private BigInteger esFalla;  
 @Column(name = "ACTIVO")  
 private BigInteger activo;  
 @Column(name = "FECHA\_REGISTRO")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaRegistro;  
 @Column(name = "USUARIO\_REGISTRO")  
 private BigInteger usuarioRegistro;  
 @Column(name = "FECHA\_MODIFICACION")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaModificacion;  
 @Column(name = "USUARIO\_MODIFICO\_ID")  
 private BigInteger usuarioModificoId;  
 @OneToMany(mappedBy = "clasificacionEventoId")  
 private Set<CatEventos> catEventosSet;  
  
 public CatClasificacionEventos() {  
 }  
  
 public CatClasificacionEventos(BigDecimal clasificacionEventoId) {  
 this.clasificacionEventoId = clasificacionEventoId;  
 }  
  
 public BigDecimal getClasificacionEventoId() {  
 return clasificacionEventoId;  
 }  
  
 public void setClasificacionEventoId(BigDecimal clasificacionEventoId) {  
 this.clasificacionEventoId = clasificacionEventoId;  
 }  
  
 public String getNombre() {  
 return nombre;  
 }  
  
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
 public String getSiglas() {  
 return siglas;  
 }  
  
 public void setSiglas(String siglas) {  
 this.siglas = siglas;  
 }  
  
 public String getDescripcion() {  
 return descripcion;  
 }  
  
 public void setDescripcion(String descripcion) {  
 this.descripcion = descripcion;  
 }  
  
 public BigInteger getRequiereOt() {  
 return requiereOt;  
 }  
  
 public void setRequiereOt(BigInteger requiereOt) {  
 this.requiereOt = requiereOt;  
 }  
  
 public BigInteger getEsFalla() {  
 return esFalla;  
 }  
  
 public void setEsFalla(BigInteger esFalla) {  
 this.esFalla = esFalla;  
 }  
  
 public BigInteger getActivo() {  
 return activo;  
 }  
  
 public void setActivo(BigInteger activo) {  
 this.activo = activo;  
 }  
  
 public Date getFechaRegistro() {  
 return fechaRegistro;  
 }  
  
 public void setFechaRegistro(Date fechaRegistro) {  
 this.fechaRegistro = fechaRegistro;  
 }  
  
 public BigInteger getUsuarioRegistro() {  
 return usuarioRegistro;  
 }  
  
 public void setUsuarioRegistro(BigInteger usuarioRegistro) {  
 this.usuarioRegistro = usuarioRegistro;  
 }  
  
 public Date getFechaModificacion() {  
 return fechaModificacion;  
 }  
  
 public void setFechaModificacion(Date fechaModificacion) {  
 this.fechaModificacion = fechaModificacion;  
 }  
  
 public BigInteger getUsuarioModificoId() {  
 return usuarioModificoId;  
 }  
  
 public void setUsuarioModificoId(BigInteger usuarioModificoId) {  
 this.usuarioModificoId = usuarioModificoId;  
 }  
  
 @XmlTransient  
 public Set<CatEventos> getCatEventosSet() {  
 return catEventosSet;  
 }  
  
 public void setCatEventosSet(Set<CatEventos> catEventosSet) {  
 this.catEventosSet = catEventosSet;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (clasificacionEventoId != null ? clasificacionEventoId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatClasificacionEventos)) {  
 return false;  
 }  
 CatClasificacionEventos other = (CatClasificacionEventos) object;  
 if ((this.clasificacionEventoId == null && other.clasificacionEventoId != null) || (this.clasificacionEventoId != null && !this.clasificacionEventoId.equals(other.clasificacionEventoId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "com.test.entity.CatClasificacionEventos[ clasificacionEventoId=" + clasificacionEventoId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatClasificacionQuery.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import java.util.Set;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlTransient;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_CLASIFICACION\_QUERY")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatClasificacionQuery.findAll", query = "SELECT c FROM CatClasificacionQuery c")  
 , @NamedQuery(name = "CatClasificacionQuery.findByClasificacionQueryId", query = "SELECT c FROM CatClasificacionQuery c WHERE c.clasificacionQueryId = :clasificacionQueryId")  
 , @NamedQuery(name = "CatClasificacionQuery.findByClasificacionQuery", query = "SELECT c FROM CatClasificacionQuery c WHERE c.clasificacionQuery = :clasificacionQuery")  
 , @NamedQuery(name = "CatClasificacionQuery.findByIdentificador", query = "SELECT c FROM CatClasificacionQuery c WHERE c.identificador = :identificador")})  
public class CatClasificacionQuery implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "CLASIFICACION\_QUERY\_ID")  
 private BigDecimal clasificacionQueryId;  
 @Size(max = 100)  
 @Column(name = "CLASIFICACION\_QUERY")  
 private String clasificacionQuery;  
 @Size(max = 3)  
 @Column(name = "IDENTIFICADOR")  
 private String identificador;  
 @OneToMany(mappedBy = "clasificacionQueryId")  
 private Set<CatTags> catTagsSet;  
  
 public CatClasificacionQuery() {  
 }  
  
 public CatClasificacionQuery(BigDecimal clasificacionQueryId) {  
 this.clasificacionQueryId = clasificacionQueryId;  
 }  
  
 public BigDecimal getClasificacionQueryId() {  
 return clasificacionQueryId;  
 }  
  
 public void setClasificacionQueryId(BigDecimal clasificacionQueryId) {  
 this.clasificacionQueryId = clasificacionQueryId;  
 }  
  
 public String getClasificacionQuery() {  
 return clasificacionQuery;  
 }  
  
 public void setClasificacionQuery(String clasificacionQuery) {  
 this.clasificacionQuery = clasificacionQuery;  
 }  
  
 public String getIdentificador() {  
 return identificador;  
 }  
  
 public void setIdentificador(String identificador) {  
 this.identificador = identificador;  
 }  
  
 @XmlTransient  
 public Set<CatTags> getCatTagsSet() {  
 return catTagsSet;  
 }  
  
 public void setCatTagsSet(Set<CatTags> catTagsSet) {  
 this.catTagsSet = catTagsSet;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (clasificacionQueryId != null ? clasificacionQueryId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatClasificacionQuery)) {  
 return false;  
 }  
 CatClasificacionQuery other = (CatClasificacionQuery) object;  
 if ((this.clasificacionQueryId == null && other.clasificacionQueryId != null) || (this.clasificacionQueryId != null && !this.clasificacionQueryId.equals(other.clasificacionQueryId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatClasificacionQuery[ clasificacionQueryId=" + clasificacionQueryId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatClienteMonedas.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatClienteMonedas generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_CLIENTE\_MONEDAS"  
)  
public class CatClienteMonedas implements java.io.Serializable {  
  
  
 private BigDecimal clienteMonedaId;  
 private CatClientes catClientes;  
 private CatMonedas catMonedas;  
  
 public CatClienteMonedas() {  
 }  
  
 public CatClienteMonedas(CatClientes catClientes, CatMonedas catMonedas) {  
 this.catClientes = catClientes;  
 this.catMonedas = catMonedas;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="CLIENTE\_MONEDA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getClienteMonedaId() {  
 return this.clienteMonedaId;  
 }  
   
 public void setClienteMonedaId(BigDecimal clienteMonedaId) {  
 this.clienteMonedaId = clienteMonedaId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="CLIENTE\_ID")  
 public CatClientes getCatClientes() {  
 return this.catClientes;  
 }  
   
 public void setCatClientes(CatClientes catClientes) {  
 this.catClientes = catClientes;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="MONEDA\_ID")  
 public CatMonedas getCatMonedas() {  
 return this.catMonedas;  
 }  
   
 public void setCatMonedas(CatMonedas catMonedas) {  
 this.catMonedas = catMonedas;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatClientes.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatClientes generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_CLIENTES"  
)  
public class CatClientes implements java.io.Serializable {  
  
  
 private BigDecimal clienteId;  
 private CatDatosFiscales catDatosFiscales;  
 private String nombreComercial;  
 private String razonSocial;  
 private String rfc;  
 private Set<CatClienteMonedas> catClienteMonedases = new HashSet<CatClienteMonedas>(0);  
 private Set<TxDocumentos> txDocumentoses = new HashSet<TxDocumentos>(0);  
  
 public CatClientes() {  
 }  
  
 public CatClientes(CatDatosFiscales catDatosFiscales, String nombreComercial, String razonSocial, String rfc, Set<CatClienteMonedas> catClienteMonedases, Set<TxDocumentos> txDocumentoses) {  
 this.catDatosFiscales = catDatosFiscales;  
 this.nombreComercial = nombreComercial;  
 this.razonSocial = razonSocial;  
 this.rfc = rfc;  
 this.catClienteMonedases = catClienteMonedases;  
 this.txDocumentoses = txDocumentoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="CLIENTE\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getClienteId() {  
 return this.clienteId;  
 }  
   
 public void setClienteId(BigDecimal clienteId) {  
 this.clienteId = clienteId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="DATOS\_FISCALES\_ID")  
 public CatDatosFiscales getCatDatosFiscales() {  
 return this.catDatosFiscales;  
 }  
   
 public void setCatDatosFiscales(CatDatosFiscales catDatosFiscales) {  
 this.catDatosFiscales = catDatosFiscales;  
 }  
  
   
 @Column(name="NOMBRE\_COMERCIAL")  
 public String getNombreComercial() {  
 return this.nombreComercial;  
 }  
   
 public void setNombreComercial(String nombreComercial) {  
 this.nombreComercial = nombreComercial;  
 }  
  
   
 @Column(name="RAZON\_SOCIAL")  
 public String getRazonSocial() {  
 return this.razonSocial;  
 }  
   
 public void setRazonSocial(String razonSocial) {  
 this.razonSocial = razonSocial;  
 }  
  
   
 @Column(name="RFC", length=20)  
 public String getRfc() {  
 return this.rfc;  
 }  
   
 public void setRfc(String rfc) {  
 this.rfc = rfc;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catClientes")  
 public Set<CatClienteMonedas> getCatClienteMonedases() {  
 return this.catClienteMonedases;  
 }  
   
 public void setCatClienteMonedases(Set<CatClienteMonedas> catClienteMonedases) {  
 this.catClienteMonedases = catClienteMonedases;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catClientes")  
 public Set<TxDocumentos> getTxDocumentoses() {  
 return this.txDocumentoses;  
 }  
   
 public void setTxDocumentoses(Set<TxDocumentos> txDocumentoses) {  
 this.txDocumentoses = txDocumentoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatCombustible.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import java.util.HashSet;  
import java.util.Set;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatCombustible generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_COMBUSTIBLE"  
)  
public class CatCombustible implements java.io.Serializable {  
  
  
 private BigDecimal combustibleId;  
 private String nombre;  
 private BigDecimal id;  
 private Set<CatLimitesDominio> catLimitesDominios = new HashSet<CatLimitesDominio>(0);  
 private Set<CatCurvasCorreccion> catCurvasCorreccions = new HashSet<CatCurvasCorreccion>(0);  
  
 public CatCombustible() {  
 }  
  
 public CatCombustible(String nombre, BigDecimal id, Set<CatLimitesDominio> catLimitesDominios, Set<CatCurvasCorreccion> catCurvasCorreccions) {  
 this.nombre = nombre;  
 this.id = id;  
 this.catLimitesDominios = catLimitesDominios;  
 this.catCurvasCorreccions = catCurvasCorreccions;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="COMBUSTIBLE\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getCombustibleId() {  
 return this.combustibleId;  
 }  
   
 public void setCombustibleId(BigDecimal combustibleId) {  
 this.combustibleId = combustibleId;  
 }  
  
   
 @Column(name="NOMBRE", length=100)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="ID", precision=22, scale=0)  
 public BigDecimal getId() {  
 return this.id;  
 }  
   
 public void setId(BigDecimal id) {  
 this.id = id;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catCombustible")  
 public Set<CatLimitesDominio> getCatLimitesDominios() {  
 return this.catLimitesDominios;  
 }  
   
 public void setCatLimitesDominios(Set<CatLimitesDominio> catLimitesDominios) {  
 this.catLimitesDominios = catLimitesDominios;  
 }  
  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catCombustible")  
 public Set<CatCurvasCorreccion> getCatCurvasCorreccions() {  
 return this.catCurvasCorreccions;  
 }  
   
 public void setCatCurvasCorreccions(Set<CatCurvasCorreccion> catCurvasCorreccions) {  
 this.catCurvasCorreccions = catCurvasCorreccions;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatCompania.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatCompania generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_COMPANIA"  
)  
public class CatCompania implements java.io.Serializable {  
  
  
 private BigDecimal companiaId;  
 private String nombreComercial;  
 private String razonSocial;  
 private String rfc;  
 private Set<CatCompaniaPlantas> catCompaniaPlantases = new HashSet<CatCompaniaPlantas>(0);  
  
 public CatCompania() {  
 }  
  
 public CatCompania(String nombreComercial, String razonSocial, String rfc, Set<CatCompaniaPlantas> catCompaniaPlantases) {  
 this.nombreComercial = nombreComercial;  
 this.razonSocial = razonSocial;  
 this.rfc = rfc;  
 this.catCompaniaPlantases = catCompaniaPlantases;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="COMPANIA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getCompaniaId() {  
 return this.companiaId;  
 }  
   
 public void setCompaniaId(BigDecimal companiaId) {  
 this.companiaId = companiaId;  
 }  
  
   
 @Column(name="NOMBRE\_COMERCIAL")  
 public String getNombreComercial() {  
 return this.nombreComercial;  
 }  
   
 public void setNombreComercial(String nombreComercial) {  
 this.nombreComercial = nombreComercial;  
 }  
  
   
 @Column(name="RAZON\_SOCIAL")  
 public String getRazonSocial() {  
 return this.razonSocial;  
 }  
   
 public void setRazonSocial(String razonSocial) {  
 this.razonSocial = razonSocial;  
 }  
  
   
 @Column(name="RFC", length=20)  
 public String getRfc() {  
 return this.rfc;  
 }  
   
 public void setRfc(String rfc) {  
 this.rfc = rfc;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catCompania")  
 public Set<CatCompaniaPlantas> getCatCompaniaPlantases() {  
 return this.catCompaniaPlantases;  
 }  
   
 public void setCatCompaniaPlantases(Set<CatCompaniaPlantas> catCompaniaPlantases) {  
 this.catCompaniaPlantases = catCompaniaPlantases;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatCompaniaPlantas.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import java.util.HashSet;  
import java.util.Set;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatCompaniaPlantas generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_COMPANIA\_PLANTAS"  
)  
public class CatCompaniaPlantas implements java.io.Serializable {  
  
  
 private BigDecimal companiaPlantaId;  
 private CatPlanta catPlanta;  
 private CatCompania catCompania;  
 private CatDatosCertificados catDatosCertificados;  
 private CatDatosFiscales catDatosFiscales;  
 private Set<TxDocumentos> txDocumentoses = new HashSet<TxDocumentos>(0);  
  
 public CatCompaniaPlantas() {  
 }  
  
 public CatCompaniaPlantas(CatPlanta catPlanta, CatCompania catCompania, CatDatosFiscales catDatosFiscales, Set<TxDocumentos> txDocumentoses) {  
 this.catPlanta = catPlanta;  
 this.catCompania = catCompania;  
 this.catDatosFiscales = catDatosFiscales;  
 this.txDocumentoses = txDocumentoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
   
 @Column(name="COMPANIA\_PLANTA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getCompaniaPlantaId() {  
 return this.companiaPlantaId;  
 }  
   
 public void setCompaniaPlantaId(BigDecimal companiaPlantaId) {  
 this.companiaPlantaId = companiaPlantaId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="PLANTA\_ID")  
 public CatPlanta getCatPlanta() {  
 return this.catPlanta;  
 }  
   
 public void setCatPlanta(CatPlanta catPlanta) {  
 this.catPlanta = catPlanta;  
 }  
  
  
 public void setCatDatosCertificados(CatDatosCertificados catDatosCertificados) {  
 this.catDatosCertificados = catDatosCertificados;  
 }  
  
 @ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="DATOS\_CERTIFICADO\_ID")  
 public CatDatosCertificados getCatDatosCertificados() {  
 return catDatosCertificados;  
 }  
  
 @ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="COMPANIA\_ID")  
 public CatCompania getCatCompania() {  
 return this.catCompania;  
 }  
   
 public void setCatCompania(CatCompania catCompania) {  
 this.catCompania = catCompania;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="DATOS\_FISCALES\_ID")  
 public CatDatosFiscales getCatDatosFiscales() {  
 return this.catDatosFiscales;  
 }  
   
 public void setCatDatosFiscales(CatDatosFiscales catDatosFiscales) {  
 this.catDatosFiscales = catDatosFiscales;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catCompaniaPlantas")  
 public Set<TxDocumentos> getTxDocumentoses() {  
 return this.txDocumentoses;  
 }  
   
 public void setTxDocumentoses(Set<TxDocumentos> txDocumentoses) {  
 this.txDocumentoses = txDocumentoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatComponentePeriodo.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatComponentePeriodo generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_COMPONENTE\_PERIODO"  
)  
public class CatComponentePeriodo implements java.io.Serializable {  
  
  
 private BigDecimal componentePeriodoId;  
 private String componentePeriodo;  
 private String formatoEstampaTiempo;  
 private Set<CatComponentes> catComponenteses = new HashSet<CatComponentes>(0);  
  
 public CatComponentePeriodo() {  
 }  
  
 public CatComponentePeriodo(String componentePeriodo, String formatoEstampaTiempo, Set<CatComponentes> catComponenteses) {  
 this.componentePeriodo = componentePeriodo;  
 this.formatoEstampaTiempo = formatoEstampaTiempo;  
 this.catComponenteses = catComponenteses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="COMPONENTE\_PERIODO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getComponentePeriodoId() {  
 return this.componentePeriodoId;  
 }  
   
 public void setComponentePeriodoId(BigDecimal componentePeriodoId) {  
 this.componentePeriodoId = componentePeriodoId;  
 }  
  
   
 @Column(name="COMPONENTE\_PERIODO", length=50)  
 public String getComponentePeriodo() {  
 return this.componentePeriodo;  
 }  
   
 public void setComponentePeriodo(String componentePeriodo) {  
 this.componentePeriodo = componentePeriodo;  
 }  
  
   
 @Column(name="FORMATO\_ESTAMPA\_TIEMPO", length=50)  
 public String getFormatoEstampaTiempo() {  
 return this.formatoEstampaTiempo;  
 }  
   
 public void setFormatoEstampaTiempo(String formatoEstampaTiempo) {  
 this.formatoEstampaTiempo = formatoEstampaTiempo;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catComponentePeriodo")  
 public Set<CatComponentes> getCatComponenteses() {  
 return this.catComponenteses;  
 }  
   
 public void setCatComponenteses(Set<CatComponentes> catComponenteses) {  
 this.catComponenteses = catComponenteses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatComponentes.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatComponentes generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_COMPONENTES"  
)  
public class CatComponentes implements java.io.Serializable {  
  
  
 private BigDecimal componenteId;  
 private CatComponentePeriodo catComponentePeriodo;  
 private String componente;  
 private String tipoComponente;  
 private Set<CatFormulaComponentes> catFormulaComponenteses = new HashSet<CatFormulaComponentes>(0);  
  
 public CatComponentes() {  
 }  
  
 public CatComponentes(CatComponentePeriodo catComponentePeriodo, String componente, String tipoComponente, Set<CatFormulaComponentes> catFormulaComponenteses) {  
 this.catComponentePeriodo = catComponentePeriodo;  
 this.componente = componente;  
 this.tipoComponente = tipoComponente;  
 this.catFormulaComponenteses = catFormulaComponenteses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="COMPONENTE\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getComponenteId() {  
 return this.componenteId;  
 }  
   
 public void setComponenteId(BigDecimal componenteId) {  
 this.componenteId = componenteId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="COMPONENTE\_PERIODO\_ID")  
 public CatComponentePeriodo getCatComponentePeriodo() {  
 return this.catComponentePeriodo;  
 }  
   
 public void setCatComponentePeriodo(CatComponentePeriodo catComponentePeriodo) {  
 this.catComponentePeriodo = catComponentePeriodo;  
 }  
  
   
 @Column(name="COMPONENTE", length=100)  
 public String getComponente() {  
 return this.componente;  
 }  
   
 public void setComponente(String componente) {  
 this.componente = componente;  
 }  
  
   
 @Column(name="TIPO\_COMPONENTE", length=100)  
 public String getTipoComponente() {  
 return this.tipoComponente;  
 }  
   
 public void setTipoComponente(String tipoComponente) {  
 this.tipoComponente = tipoComponente;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catComponentes")  
 public Set<CatFormulaComponentes> getCatFormulaComponenteses() {  
 return this.catFormulaComponenteses;  
 }  
   
 public void setCatFormulaComponenteses(Set<CatFormulaComponentes> catFormulaComponenteses) {  
 this.catFormulaComponenteses = catFormulaComponenteses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatCondiciones.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatCondiciones generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_CONDICIONES"  
)  
public class CatCondiciones implements java.io.Serializable {  
  
  
 private BigDecimal condicionId;  
 private CatPlanta catPlanta;  
 private String proceso;  
 private String keyCondicion;  
 private String valueCondicion;  
  
 public CatCondiciones() {  
 }  
  
 public CatCondiciones(CatPlanta catPlanta, String keyCondicion, String valueCondicion) {  
 this.catPlanta = catPlanta;  
 this.setKeyCondicion(keyCondicion);  
 this.setValueCondicion(valueCondicion);  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
 @Column(name="CONDICION\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getCondicionId() {  
 return this.condicionId;  
 }  
   
 public void setCondicionId(BigDecimal condicionId) {  
 this.condicionId = condicionId;  
 }  
  
 @ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="PLANTA\_ID", nullable=false)  
 public CatPlanta getCatPlanta() {  
 return this.catPlanta;  
 }  
   
 public void setCatPlanta(CatPlanta catPlanta) {  
 this.catPlanta = catPlanta;  
 }  
   
 public String getKeyCondicion() {  
 return keyCondicion;  
 }  
  
 public void setKeyCondicion(String keyCondicion) {  
 this.keyCondicion = keyCondicion;  
 }  
  
 public String getValueCondicion() {  
 return valueCondicion;  
 }  
  
 public void setValueCondicion(String valueCondicion) {  
 this.valueCondicion = valueCondicion;  
 }  
  
 public String getProceso() {  
 return proceso;  
 }  
  
 public void setProceso(String proceso) {  
 this.proceso = proceso;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatConexiones.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.NamedStoredProcedureQueries;  
import javax.persistence.NamedStoredProcedureQuery;  
import javax.persistence.ParameterMode;  
import javax.persistence.StoredProcedureParameter;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatConexiones generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_CONEXIONES")  
@NamedStoredProcedureQueries({  
 @NamedStoredProcedureQuery(  
 name = "procesaFileSol",   
 procedureName = "QADGC.DGC.LOAD\_SOL",  
 parameters = {  
 @StoredProcedureParameter(mode = ParameterMode.IN, type = String.class, name = "P\_TABLE"),   
 @StoredProcedureParameter(mode = ParameterMode.IN, type = String.class, name = "P\_DIR"),   
 @StoredProcedureParameter(mode = ParameterMode.IN, type = String.class, name = "P\_FILENAME"),   
 @StoredProcedureParameter(mode = ParameterMode.IN, type = Integer.class, name = "P\_IGNORE\_HEADERLINES"),  
 @StoredProcedureParameter(mode = ParameterMode.IN, type = Integer.class, name = "P\_TYPE\_LOAD"),  
 @StoredProcedureParameter(mode = ParameterMode.IN, type = Integer.class, name = "P\_ID\_USUARIO")  
 }  
),  
 @NamedStoredProcedureQuery(  
 name = "procesaFileAguila",   
 procedureName = "QADGC.DGC.LOAD\_AGUILA",  
 parameters = {  
 @StoredProcedureParameter(mode = ParameterMode.IN, type = String.class, name = "P\_TABLE"),   
 @StoredProcedureParameter(mode = ParameterMode.IN, type = String.class, name = "P\_DIR"),   
 @StoredProcedureParameter(mode = ParameterMode.IN, type = String.class, name = "P\_FILENAME"),   
 @StoredProcedureParameter(mode = ParameterMode.IN, type = Integer.class, name = "P\_IGNORE\_HEADERLINES"),  
 @StoredProcedureParameter(mode = ParameterMode.IN, type = Integer.class, name = "P\_TYPE\_LOAD"),  
 @StoredProcedureParameter(mode = ParameterMode.IN, type = Integer.class, name = "P\_ID\_USUARIO")  
 }  
)  
})  
  
public class CatConexiones implements java.io.Serializable {  
  
  
 private BigDecimal conexionId;  
 private String conexion;  
 private String host;  
 private String puerto;  
 private String username;  
 private String password;  
 private String homePath;  
 private String pathRemoto;  
  
 public CatConexiones() {  
 }  
  
 public CatConexiones(String conexion, String host, String puerto, String username, String password, String homePath, String pathRemoto) {  
 this.conexion = conexion;  
 this.host = host;  
 this.puerto = puerto;  
 this.username = username;  
 this.password = password;  
 this.homePath = homePath;  
 this.pathRemoto = pathRemoto;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="CONEXION\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getConexionId() {  
 return this.conexionId;  
 }  
   
 public void setConexionId(BigDecimal conexionId) {  
 this.conexionId = conexionId;  
 }  
  
   
 @Column(name="CONEXION", length=200)  
 public String getConexion() {  
 return this.conexion;  
 }  
   
 public void setConexion(String conexion) {  
 this.conexion = conexion;  
 }  
  
   
 @Column(name="HOST", length=200)  
 public String getHost() {  
 return this.host;  
 }  
   
 public void setHost(String host) {  
 this.host = host;  
 }  
  
   
 @Column(name="PUERTO", length=20)  
 public String getPuerto() {  
 return this.puerto;  
 }  
   
 public void setPuerto(String puerto) {  
 this.puerto = puerto;  
 }  
  
   
 @Column(name="USERNAME", length=200)  
 public String getUsername() {  
 return this.username;  
 }  
   
 public void setUsername(String username) {  
 this.username = username;  
 }  
  
   
 @Column(name="PASSWORD", length=200)  
 public String getPassword() {  
 return this.password;  
 }  
   
 public void setPassword(String password) {  
 this.password = password;  
 }  
  
   
 @Column(name="HOME\_PATH", length=200)  
 public String getHomePath() {  
 return this.homePath;  
 }  
   
 public void setHomePath(String homePath) {  
 this.homePath = homePath;  
 }  
  
   
 @Column(name="PATH\_REMOTO", length=200)  
 public String getPathRemoto() {  
 return this.pathRemoto;  
 }  
   
 public void setPathRemoto(String pathRemoto) {  
 this.pathRemoto = pathRemoto;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatConfiguracionDetalle.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import java.util.Set;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.validation.constraints.NotNull;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlTransient;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_CONFIGURACION\_DETALLE")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatConfiguracionDetalle.findAll", query = "SELECT c FROM CatConfiguracionDetalle c")  
 , @NamedQuery(name = "CatConfiguracionDetalle.findByConfiguracionDetalleId", query = "SELECT c FROM CatConfiguracionDetalle c WHERE c.configuracionDetalleId = :configuracionDetalleId")})  
public class CatConfiguracionDetalle implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "CONFIGURACION\_DETALLE\_ID")  
 private BigDecimal configuracionDetalleId;  
 @JoinColumn(name = "CONFIGURACION\_EVENTO\_ID", referencedColumnName = "CONFIGURACION\_EVENTO\_ID")  
 @ManyToOne  
 private CatConfiguracionEventos configuracionEventoId;  
 @JoinColumn(name = "DETALLE\_ID", referencedColumnName = "DETALLE\_ID")  
 @ManyToOne  
 private CatDetalles detalleId;  
 @OneToMany(mappedBy = "configuracionDetalleId")  
 private Set<CatConfiguracionOpciones> catConfiguracionOpcionesSet;  
  
 public CatConfiguracionDetalle() {  
 }  
  
 public CatConfiguracionDetalle(BigDecimal configuracionDetalleId) {  
 this.configuracionDetalleId = configuracionDetalleId;  
 }  
  
 public BigDecimal getConfiguracionDetalleId() {  
 return configuracionDetalleId;  
 }  
  
 public void setConfiguracionDetalleId(BigDecimal configuracionDetalleId) {  
 this.configuracionDetalleId = configuracionDetalleId;  
 }  
  
 public CatConfiguracionEventos getConfiguracionEventoId() {  
 return configuracionEventoId;  
 }  
  
 public void setConfiguracionEventoId(CatConfiguracionEventos configuracionEventoId) {  
 this.configuracionEventoId = configuracionEventoId;  
 }  
  
 public CatDetalles getDetalleId() {  
 return detalleId;  
 }  
  
 public void setDetalleId(CatDetalles detalleId) {  
 this.detalleId = detalleId;  
 }  
  
 @XmlTransient  
 public Set<CatConfiguracionOpciones> getCatConfiguracionOpcionesSet() {  
 return catConfiguracionOpcionesSet;  
 }  
  
 public void setCatConfiguracionOpcionesSet(Set<CatConfiguracionOpciones> catConfiguracionOpcionesSet) {  
 this.catConfiguracionOpcionesSet = catConfiguracionOpcionesSet;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (configuracionDetalleId != null ? configuracionDetalleId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatConfiguracionDetalle)) {  
 return false;  
 }  
 CatConfiguracionDetalle other = (CatConfiguracionDetalle) object;  
 if ((this.configuracionDetalleId == null && other.configuracionDetalleId != null) || (this.configuracionDetalleId != null && !this.configuracionDetalleId.equals(other.configuracionDetalleId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatConfiguracionDetalle[ configuracionDetalleId=" + configuracionDetalleId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatConfiguracionEventos.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import java.util.Set;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlTransient;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_CONFIGURACION\_EVENTOS")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatConfiguracionEventos.findAll", query = "SELECT c FROM CatConfiguracionEventos c")  
 , @NamedQuery(name = "CatConfiguracionEventos.findByConfiguracionEventoId", query = "SELECT c FROM CatConfiguracionEventos c WHERE c.configuracionEventoId = :configuracionEventoId")  
 , @NamedQuery(name = "CatConfiguracionEventos.findByConfiguracion", query = "SELECT c FROM CatConfiguracionEventos c WHERE c.configuracion = :configuracion")  
 , @NamedQuery(name = "CatConfiguracionEventos.findByEsFalla", query = "SELECT c FROM CatConfiguracionEventos c WHERE c.esFalla = :esFalla")  
 , @NamedQuery(name = "CatConfiguracionEventos.findByActivo", query = "SELECT c FROM CatConfiguracionEventos c WHERE c.activo = :activo")})  
public class CatConfiguracionEventos implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "CONFIGURACION\_EVENTO\_ID")  
 private BigDecimal configuracionEventoId;  
 @Size(max = 250)  
 @Column(name = "CONFIGURACION")  
 private String configuracion;  
 @Column(name = "ES\_FALLA")  
 private Short esFalla;  
 @Column(name = "ACTIVO")  
 private Short activo;  
 @OneToMany(mappedBy = "configuracionEventoId")  
 private Set<CatConfiguracionDetalle> catConfiguracionDetalleSet;  
 @JoinColumn(name = "CLASIFICACION\_EVENTO\_ID", referencedColumnName = "CLASIFICACION\_EVENTO\_ID")  
 @ManyToOne  
 private CatClasificacionEvento clasificacionEventoId;  
  
 public CatConfiguracionEventos() {  
 }  
  
 public CatConfiguracionEventos(BigDecimal configuracionEventoId) {  
 this.configuracionEventoId = configuracionEventoId;  
 }  
  
 public BigDecimal getConfiguracionEventoId() {  
 return configuracionEventoId;  
 }  
  
 public void setConfiguracionEventoId(BigDecimal configuracionEventoId) {  
 this.configuracionEventoId = configuracionEventoId;  
 }  
  
 public String getConfiguracion() {  
 return configuracion;  
 }  
  
 public void setConfiguracion(String configuracion) {  
 this.configuracion = configuracion;  
 }  
  
 public Short getEsFalla() {  
 return esFalla;  
 }  
  
 public void setEsFalla(Short esFalla) {  
 this.esFalla = esFalla;  
 }  
  
 public Short getActivo() {  
 return activo;  
 }  
  
 public void setActivo(Short activo) {  
 this.activo = activo;  
 }  
  
 @XmlTransient  
 public Set<CatConfiguracionDetalle> getCatConfiguracionDetalleSet() {  
 return catConfiguracionDetalleSet;  
 }  
  
 public void setCatConfiguracionDetalleSet(Set<CatConfiguracionDetalle> catConfiguracionDetalleSet) {  
 this.catConfiguracionDetalleSet = catConfiguracionDetalleSet;  
 }  
  
 public CatClasificacionEvento getClasificacionEventoId() {  
 return clasificacionEventoId;  
 }  
  
 public void setClasificacionEventoId(CatClasificacionEvento clasificacionEventoId) {  
 this.clasificacionEventoId = clasificacionEventoId;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (configuracionEventoId != null ? configuracionEventoId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatConfiguracionEventos)) {  
 return false;  
 }  
 CatConfiguracionEventos other = (CatConfiguracionEventos) object;  
 if ((this.configuracionEventoId == null && other.configuracionEventoId != null) || (this.configuracionEventoId != null && !this.configuracionEventoId.equals(other.configuracionEventoId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatConfiguracionEventos[ configuracionEventoId=" + configuracionEventoId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatConfiguracionOpciones.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.Table;  
import javax.validation.constraints.NotNull;  
import javax.xml.bind.annotation.XmlRootElement;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_CONFIGURACION\_OPCIONES")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatConfiguracionOpciones.findAll", query = "SELECT c FROM CatConfiguracionOpciones c")  
 , @NamedQuery(name = "CatConfiguracionOpciones.findByConfiguracionOpcionesId", query = "SELECT c FROM CatConfiguracionOpciones c WHERE c.configuracionOpcionesId = :configuracionOpcionesId")})  
public class CatConfiguracionOpciones implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "CONFIGURACION\_OPCIONES\_ID")  
 private BigDecimal configuracionOpcionesId;  
 @JoinColumn(name = "CONFIGURACION\_DETALLE\_ID", referencedColumnName = "CONFIGURACION\_DETALLE\_ID")  
 @ManyToOne  
 private CatConfiguracionDetalle configuracionDetalleId;  
 @JoinColumn(name = "DETALLE\_OPCIONES\_ID", referencedColumnName = "DETALLE\_OPCIONES\_ID")  
 @ManyToOne  
 private CatDetalleOpciones detalleOpcionesId;  
  
 public CatConfiguracionOpciones() {  
 }  
  
 public CatConfiguracionOpciones(BigDecimal configuracionOpcionesId) {  
 this.configuracionOpcionesId = configuracionOpcionesId;  
 }  
  
 public BigDecimal getConfiguracionOpcionesId() {  
 return configuracionOpcionesId;  
 }  
  
 public void setConfiguracionOpcionesId(BigDecimal configuracionOpcionesId) {  
 this.configuracionOpcionesId = configuracionOpcionesId;  
 }  
  
 public CatConfiguracionDetalle getConfiguracionDetalleId() {  
 return configuracionDetalleId;  
 }  
  
 public void setConfiguracionDetalleId(CatConfiguracionDetalle configuracionDetalleId) {  
 this.configuracionDetalleId = configuracionDetalleId;  
 }  
  
 public CatDetalleOpciones getDetalleOpcionesId() {  
 return detalleOpcionesId;  
 }  
  
 public void setDetalleOpcionesId(CatDetalleOpciones detalleOpcionesId) {  
 this.detalleOpcionesId = detalleOpcionesId;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (configuracionOpcionesId != null ? configuracionOpcionesId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatConfiguracionOpciones)) {  
 return false;  
 }  
 CatConfiguracionOpciones other = (CatConfiguracionOpciones) object;  
 if ((this.configuracionOpcionesId == null && other.configuracionOpcionesId != null) || (this.configuracionOpcionesId != null && !this.configuracionOpcionesId.equals(other.configuracionOpcionesId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatConfiguracionOpciones[ configuracionOpcionesId=" + configuracionOpcionesId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatContratosAfectados.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import java.math.BigInteger;  
import java.util.Date;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
import javax.xml.bind.annotation.XmlRootElement;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_CONTRATOS\_AFECTADOS")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatContratosAfectados.findAll", query = "SELECT c FROM CatContratosAfectados c")  
 , @NamedQuery(name = "CatContratosAfectados.findByContratoAfectadoId", query = "SELECT c FROM CatContratosAfectados c WHERE c.contratoAfectadoId = :contratoAfectadoId")  
 , @NamedQuery(name = "CatContratosAfectados.findByNombre", query = "SELECT c FROM CatContratosAfectados c WHERE c.nombre = :nombre")  
 , @NamedQuery(name = "CatContratosAfectados.findBySiglas", query = "SELECT c FROM CatContratosAfectados c WHERE c.siglas = :siglas")  
 , @NamedQuery(name = "CatContratosAfectados.findByVisibleEnBitacora", query = "SELECT c FROM CatContratosAfectados c WHERE c.visibleEnBitacora = :visibleEnBitacora")  
 , @NamedQuery(name = "CatContratosAfectados.findByDescripcion", query = "SELECT c FROM CatContratosAfectados c WHERE c.descripcion = :descripcion")  
 , @NamedQuery(name = "CatContratosAfectados.findByMemRequiereId", query = "SELECT c FROM CatContratosAfectados c WHERE c.memRequiereId = :memRequiereId")  
 , @NamedQuery(name = "CatContratosAfectados.findByActivo", query = "SELECT c FROM CatContratosAfectados c WHERE c.activo = :activo")  
 , @NamedQuery(name = "CatContratosAfectados.findByFechaRegistro", query = "SELECT c FROM CatContratosAfectados c WHERE c.fechaRegistro = :fechaRegistro")  
 , @NamedQuery(name = "CatContratosAfectados.findByUsuarioRegistro", query = "SELECT c FROM CatContratosAfectados c WHERE c.usuarioRegistro = :usuarioRegistro")  
 , @NamedQuery(name = "CatContratosAfectados.findByFechaModificacion", query = "SELECT c FROM CatContratosAfectados c WHERE c.fechaModificacion = :fechaModificacion")  
 , @NamedQuery(name = "CatContratosAfectados.findByUsuarioModificoId", query = "SELECT c FROM CatContratosAfectados c WHERE c.usuarioModificoId = :usuarioModificoId")})  
public class CatContratosAfectados implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "CONTRATO\_AFECTADO\_ID")  
 private BigDecimal contratoAfectadoId;  
 @Size(max = 1020)  
 @Column(name = "NOMBRE")  
 private String nombre;  
 @Size(max = 4000)  
 @Column(name = "SIGLAS")  
 private String siglas;  
 @Column(name = "VISIBLE\_EN\_BITACORA")  
 private BigInteger visibleEnBitacora;  
 @Size(max = 4000)  
 @Column(name = "DESCRIPCION")  
 private String descripcion;  
 @Column(name = "MEM\_REQUIERE\_ID")  
 private BigInteger memRequiereId;  
 @Column(name = "ACTIVO")  
 private BigInteger activo;  
 @Column(name = "FECHA\_REGISTRO")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaRegistro;  
 @Column(name = "USUARIO\_REGISTRO")  
 private BigInteger usuarioRegistro;  
 @Column(name = "FECHA\_MODIFICACION")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaModificacion;  
 @Column(name = "USUARIO\_MODIFICO\_ID")  
 private BigInteger usuarioModificoId;  
  
 public CatContratosAfectados() {  
 }  
  
 public CatContratosAfectados(BigDecimal contratoAfectadoId) {  
 this.contratoAfectadoId = contratoAfectadoId;  
 }  
  
 public BigDecimal getContratoAfectadoId() {  
 return contratoAfectadoId;  
 }  
  
 public void setContratoAfectadoId(BigDecimal contratoAfectadoId) {  
 this.contratoAfectadoId = contratoAfectadoId;  
 }  
  
 public String getNombre() {  
 return nombre;  
 }  
  
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
 public String getSiglas() {  
 return siglas;  
 }  
  
 public void setSiglas(String siglas) {  
 this.siglas = siglas;  
 }  
  
 public BigInteger getVisibleEnBitacora() {  
 return visibleEnBitacora;  
 }  
  
 public void setVisibleEnBitacora(BigInteger visibleEnBitacora) {  
 this.visibleEnBitacora = visibleEnBitacora;  
 }  
  
 public String getDescripcion() {  
 return descripcion;  
 }  
  
 public void setDescripcion(String descripcion) {  
 this.descripcion = descripcion;  
 }  
  
 public BigInteger getMemRequiereId() {  
 return memRequiereId;  
 }  
  
 public void setMemRequiereId(BigInteger memRequiereId) {  
 this.memRequiereId = memRequiereId;  
 }  
  
 public BigInteger getActivo() {  
 return activo;  
 }  
  
 public void setActivo(BigInteger activo) {  
 this.activo = activo;  
 }  
  
 public Date getFechaRegistro() {  
 return fechaRegistro;  
 }  
  
 public void setFechaRegistro(Date fechaRegistro) {  
 this.fechaRegistro = fechaRegistro;  
 }  
  
 public BigInteger getUsuarioRegistro() {  
 return usuarioRegistro;  
 }  
  
 public void setUsuarioRegistro(BigInteger usuarioRegistro) {  
 this.usuarioRegistro = usuarioRegistro;  
 }  
  
 public Date getFechaModificacion() {  
 return fechaModificacion;  
 }  
  
 public void setFechaModificacion(Date fechaModificacion) {  
 this.fechaModificacion = fechaModificacion;  
 }  
  
 public BigInteger getUsuarioModificoId() {  
 return usuarioModificoId;  
 }  
  
 public void setUsuarioModificoId(BigInteger usuarioModificoId) {  
 this.usuarioModificoId = usuarioModificoId;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (contratoAfectadoId != null ? contratoAfectadoId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatContratosAfectados)) {  
 return false;  
 }  
 CatContratosAfectados other = (CatContratosAfectados) object;  
 if ((this.contratoAfectadoId == null && other.contratoAfectadoId != null) || (this.contratoAfectadoId != null && !this.contratoAfectadoId.equals(other.contratoAfectadoId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "com.test.entity.CatContratosAfectados[ contratoAfectadoId=" + contratoAfectadoId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatCurvasCorreccion.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatCurvasCorreccion generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_CURVAS\_CORRECCION"  
)  
public class CatCurvasCorreccion implements java.io.Serializable {  
  
  
 private BigDecimal curvaId;  
 private CatCapacidad catCapacidad;  
 private CatCombustible catCombustible;  
 private CatFactorTipo catFactorTipo;  
 private CatPlanta catPlanta;  
 private CatVariable catVariable;  
 private BigDecimal x1;  
 private BigDecimal x2;  
 private String fx;  
  
 public CatCurvasCorreccion() {  
 }  
  
   
 public CatCurvasCorreccion(CatCapacidad catCapacidad, CatPlanta catPlanta, CatVariable catVariable) {  
 this.catCapacidad = catCapacidad;  
 this.catPlanta = catPlanta;  
 this.catVariable = catVariable;  
 }  
 public CatCurvasCorreccion(CatCapacidad catCapacidad, CatCombustible catCombustible, CatFactorTipo catFactorTipo, CatPlanta catPlanta, CatVariable catVariable, BigDecimal x1, BigDecimal x2, String fx) {  
 this.catCapacidad = catCapacidad;  
 this.catCombustible = catCombustible;  
 this.catFactorTipo = catFactorTipo;  
 this.catPlanta = catPlanta;  
 this.catVariable = catVariable;  
 this.x1 = x1;  
 this.x2 = x2;  
 this.fx = fx;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="CURVA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getCurvaId() {  
 return this.curvaId;  
 }  
   
 public void setCurvaId(BigDecimal curvaId) {  
 this.curvaId = curvaId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="CAPACIDAD\_ID", nullable=false)  
 public CatCapacidad getCatCapacidad() {  
 return this.catCapacidad;  
 }  
   
 public void setCatCapacidad(CatCapacidad catCapacidad) {  
 this.catCapacidad = catCapacidad;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="COMBUSTIBLE\_ID")  
 public CatCombustible getCatCombustible() {  
 return this.catCombustible;  
 }  
   
 public void setCatCombustible(CatCombustible catCombustible) {  
 this.catCombustible = catCombustible;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="FACTOR\_TIPO\_ID")  
 public CatFactorTipo getCatFactorTipo() {  
 return this.catFactorTipo;  
 }  
   
 public void setCatFactorTipo(CatFactorTipo catFactorTipo) {  
 this.catFactorTipo = catFactorTipo;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="PLANTA\_ID", nullable=false)  
 public CatPlanta getCatPlanta() {  
 return this.catPlanta;  
 }  
   
 public void setCatPlanta(CatPlanta catPlanta) {  
 this.catPlanta = catPlanta;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="VARIABLE\_ID", nullable=false)  
 public CatVariable getCatVariable() {  
 return this.catVariable;  
 }  
   
 public void setCatVariable(CatVariable catVariable) {  
 this.catVariable = catVariable;  
 }  
  
   
 @Column(name="X1", precision=22, scale=0)  
 public BigDecimal getX1() {  
 return this.x1;  
 }  
   
 public void setX1(BigDecimal x1) {  
 this.x1 = x1;  
 }  
  
   
 @Column(name="X2", precision=22, scale=0)  
 public BigDecimal getX2() {  
 return this.x2;  
 }  
   
 public void setX2(BigDecimal x2) {  
 this.x2 = x2;  
 }  
  
   
 @Column(name="FX", length=1000)  
 public String getFx() {  
 return this.fx;  
 }  
   
 public void setFx(String fx) {  
 this.fx = fx;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatDatosCertificados.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.Table;  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
import javax.xml.bind.annotation.XmlRootElement;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_DATOS\_CERTIFICADOS")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatDatosCertificados.findAll", query = "SELECT c FROM CatDatosCertificados c")  
 , @NamedQuery(name = "CatDatosCertificados.findByDatosCertificadoId", query = "SELECT c FROM CatDatosCertificados c WHERE c.datosCertificadoId = :datosCertificadoId")  
 , @NamedQuery(name = "CatDatosCertificados.findByNumeroCertificado", query = "SELECT c FROM CatDatosCertificados c WHERE c.numeroCertificado = :numeroCertificado")  
 , @NamedQuery(name = "CatDatosCertificados.findByArchivoCertificado", query = "SELECT c FROM CatDatosCertificados c WHERE c.archivoCertificado = :archivoCertificado")  
 , @NamedQuery(name = "CatDatosCertificados.findByArchivoKey", query = "SELECT c FROM CatDatosCertificados c WHERE c.archivoKey = :archivoKey")  
 , @NamedQuery(name = "CatDatosCertificados.findByContrasenaKey", query = "SELECT c FROM CatDatosCertificados c WHERE c.contrasenaKey = :contrasenaKey")})  
public class CatDatosCertificados implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "DATOS\_CERTIFICADO\_ID")  
 private BigDecimal datosCertificadoId;  
 @Size(max = 20)  
 @Column(name = "NUMERO\_CERTIFICADO")  
 private String numeroCertificado;  
 @Size(max = 255)  
 @Column(name = "ARCHIVO\_CERTIFICADO")  
 private String archivoCertificado;  
 @Size(max = 255)  
 @Column(name = "ARCHIVO\_KEY")  
 private String archivoKey;  
 @Size(max = 255)  
 @Column(name = "CONTRASENA\_KEY")  
 private String contrasenaKey;  
  
 public CatDatosCertificados() {  
 }  
  
 public CatDatosCertificados(BigDecimal datosCertificadoId) {  
 this.datosCertificadoId = datosCertificadoId;  
 }  
  
 public BigDecimal getDatosCertificadoId() {  
 return datosCertificadoId;  
 }  
  
 public void setDatosCertificadoId(BigDecimal datosCertificadoId) {  
 this.datosCertificadoId = datosCertificadoId;  
 }  
  
 public String getNumeroCertificado() {  
 return numeroCertificado;  
 }  
  
 public void setNumeroCertificado(String numeroCertificado) {  
 this.numeroCertificado = numeroCertificado;  
 }  
  
 public String getArchivoCertificado() {  
 return archivoCertificado;  
 }  
  
 public void setArchivoCertificado(String archivoCertificado) {  
 this.archivoCertificado = archivoCertificado;  
 }  
  
 public String getArchivoKey() {  
 return archivoKey;  
 }  
  
 public void setArchivoKey(String archivoKey) {  
 this.archivoKey = archivoKey;  
 }  
  
 public String getContrasenaKey() {  
 return contrasenaKey;  
 }  
  
 public void setContrasenaKey(String contrasenaKey) {  
 this.contrasenaKey = contrasenaKey;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (datosCertificadoId != null ? datosCertificadoId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatDatosCertificados)) {  
 return false;  
 }  
 CatDatosCertificados other = (CatDatosCertificados) object;  
 if ((this.datosCertificadoId == null && other.datosCertificadoId != null) || (this.datosCertificadoId != null && !this.datosCertificadoId.equals(other.datosCertificadoId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatDatosCertificados[ datosCertificadoId=" + datosCertificadoId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatDatosFiscales.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatDatosFiscales generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_DATOS\_FISCALES"  
)  
public class CatDatosFiscales implements java.io.Serializable {  
  
  
 private BigDecimal datosFiscalesId;  
 private CatEstados catEstados;  
 private CatRegimenFiscal catRegimenFiscal;  
 private String calle;  
 private String numExt;  
 private String numInt;  
 private String colonia;  
 private String delegacionOMunicipio;  
 private BigDecimal cp;  
 private Boolean principal;  
 private String registroIdTributario;  
 private Set<CatCompaniaPlantas> catCompaniaPlantases = new HashSet<CatCompaniaPlantas>(0);  
 private Set<CatClientes> catClienteses = new HashSet<CatClientes>(0);  
  
 public CatDatosFiscales() {  
 }  
  
 public CatDatosFiscales(CatEstados catEstados, CatRegimenFiscal catRegimenFiscal, String calle, String numExt, String numInt, String colonia, String delegacionOMunicipio, BigDecimal cp, Boolean principal, String registroIdTributario, Set<CatCompaniaPlantas> catCompaniaPlantases, Set<CatClientes> catClienteses) {  
 this.catEstados = catEstados;  
 this.catRegimenFiscal = catRegimenFiscal;  
 this.calle = calle;  
 this.numExt = numExt;  
 this.numInt = numInt;  
 this.colonia = colonia;  
 this.delegacionOMunicipio = delegacionOMunicipio;  
 this.cp = cp;  
 this.principal = principal;  
 this.registroIdTributario = registroIdTributario;  
 this.catCompaniaPlantases = catCompaniaPlantases;  
 this.catClienteses = catClienteses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="DATOS\_FISCALES\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getDatosFiscalesId() {  
 return this.datosFiscalesId;  
 }  
   
 public void setDatosFiscalesId(BigDecimal datosFiscalesId) {  
 this.datosFiscalesId = datosFiscalesId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="ESTADO\_ID")  
 public CatEstados getCatEstados() {  
 return this.catEstados;  
 }  
   
 public void setCatEstados(CatEstados catEstados) {  
 this.catEstados = catEstados;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="REGIMEN\_FISCAL\_ID")  
 public CatRegimenFiscal getCatRegimenFiscal() {  
 return this.catRegimenFiscal;  
 }  
   
 public void setCatRegimenFiscal(CatRegimenFiscal catRegimenFiscal) {  
 this.catRegimenFiscal = catRegimenFiscal;  
 }  
  
   
 @Column(name="CALLE")  
 public String getCalle() {  
 return this.calle;  
 }  
   
 public void setCalle(String calle) {  
 this.calle = calle;  
 }  
  
   
 @Column(name="NUM\_EXT")  
 public String getNumExt() {  
 return this.numExt;  
 }  
   
 public void setNumExt(String numExt) {  
 this.numExt = numExt;  
 }  
  
   
 @Column(name="NUM\_INT")  
 public String getNumInt() {  
 return this.numInt;  
 }  
   
 public void setNumInt(String numInt) {  
 this.numInt = numInt;  
 }  
  
   
 @Column(name="COLONIA")  
 public String getColonia() {  
 return this.colonia;  
 }  
   
 public void setColonia(String colonia) {  
 this.colonia = colonia;  
 }  
  
   
 @Column(name="DELEGACION\_O\_MUNICIPIO")  
 public String getDelegacionOMunicipio() {  
 return this.delegacionOMunicipio;  
 }  
   
 public void setDelegacionOMunicipio(String delegacionOMunicipio) {  
 this.delegacionOMunicipio = delegacionOMunicipio;  
 }  
  
   
 @Column(name="CP", precision=22, scale=0)  
 public BigDecimal getCp() {  
 return this.cp;  
 }  
   
 public void setCp(BigDecimal cp) {  
 this.cp = cp;  
 }  
  
   
 @Column(name="PRINCIPAL", precision=1, scale=0)  
 public Boolean getPrincipal() {  
 return this.principal;  
 }  
   
 public void setPrincipal(Boolean principal) {  
 this.principal = principal;  
 }  
  
   
 @Column(name="REGISTRO\_ID\_TRIBUTARIO")  
 public String getRegistroIdTributario() {  
 return this.registroIdTributario;  
 }  
   
 public void setRegistroIdTributario(String registroIdTributario) {  
 this.registroIdTributario = registroIdTributario;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catDatosFiscales")  
 public Set<CatCompaniaPlantas> getCatCompaniaPlantases() {  
 return this.catCompaniaPlantases;  
 }  
   
 public void setCatCompaniaPlantases(Set<CatCompaniaPlantas> catCompaniaPlantases) {  
 this.catCompaniaPlantases = catCompaniaPlantases;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catDatosFiscales")  
 public Set<CatClientes> getCatClienteses() {  
 return this.catClienteses;  
 }  
   
 public void setCatClienteses(Set<CatClientes> catClienteses) {  
 this.catClienteses = catClienteses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatDetalleOpciones.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
  
import java.util.HashMap;  
import java.util.Set;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlTransient;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_DETALLE\_OPCIONES")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatDetalleOpciones.findAll", query = "SELECT c FROM CatDetalleOpciones c")  
 , @NamedQuery(name = "CatDetalleOpciones.findByDetalleOpcionesId", query = "SELECT c FROM CatDetalleOpciones c WHERE c.detalleOpcionesId = :detalleOpcionesId")  
 , @NamedQuery(name = "CatDetalleOpciones.findByOpcion", query = "SELECT c FROM CatDetalleOpciones c WHERE c.opcion = :opcion")  
 , @NamedQuery(name = "CatDetalleOpciones.findByActivo", query = "SELECT c FROM CatDetalleOpciones c WHERE c.activo = :activo")})  
public class CatDetalleOpciones implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "DETALLE\_OPCIONES\_ID")  
 private BigDecimal detalleOpcionesId;  
 @Size(max = 250)  
 @Column(name = "OPCION")  
 private String opcion;  
 @Column(name = "ACTIVO")  
 private Short activo;  
 @OneToMany(mappedBy = "detalleOpcionesId")  
 private Set<CatOpcionPlanta> catOpcionPlantaSet;  
 @OneToMany(mappedBy = "detalleOpcionesId")  
 private Set<CatConfiguracionOpciones> catConfiguracionOpcionesSet;  
 @OneToMany(mappedBy = "detalleOpcionesId")  
 private Set<CatOpcionTag> catOpcionTagSet;  
 @JoinColumn(name = "DETALLE\_ID", referencedColumnName = "DETALLE\_ID")  
 @ManyToOne  
 private CatDetalles detalleId;  
 @OneToMany(mappedBy = "detalleOpcionesId")  
 private Set<CatOpcionTraduccion> catOpcionTraduccionSet;  
  
 public CatDetalleOpciones() {  
 }  
  
 public CatDetalleOpciones(BigDecimal detalleOpcionesId) {  
 this.detalleOpcionesId = detalleOpcionesId;  
 }  
  
 public BigDecimal getDetalleOpcionesId() {  
 return detalleOpcionesId;  
 }  
  
 public void setDetalleOpcionesId(BigDecimal detalleOpcionesId) {  
 this.detalleOpcionesId = detalleOpcionesId;  
 }  
  
 public String getOpcion() {  
 return opcion;  
 }  
  
 public void setOpcion(String opcion) {  
 this.opcion = opcion;  
 }  
  
 public Short getActivo() {  
 return activo;  
 }  
  
 public void setActivo(Short activo) {  
 this.activo = activo;  
 }  
  
 @XmlTransient  
 public Set<CatOpcionPlanta> getCatOpcionPlantaSet() {  
 return catOpcionPlantaSet;  
 }  
  
 public void setCatOpcionPlantaSet(Set<CatOpcionPlanta> catOpcionPlantaSet) {  
 this.catOpcionPlantaSet = catOpcionPlantaSet;  
 }  
  
 @XmlTransient  
 public Set<CatConfiguracionOpciones> getCatConfiguracionOpcionesSet() {  
 return catConfiguracionOpcionesSet;  
 }  
  
 public void setCatConfiguracionOpcionesSet(Set<CatConfiguracionOpciones> catConfiguracionOpcionesSet) {  
 this.catConfiguracionOpcionesSet = catConfiguracionOpcionesSet;  
 }  
  
 @XmlTransient  
 public Set<CatOpcionTag> getCatOpcionTagSet() {  
 return catOpcionTagSet;  
 }  
  
 public void setCatOpcionTagSet(Set<CatOpcionTag> catOpcionTagSet) {  
 this.catOpcionTagSet = catOpcionTagSet;  
 }  
  
 public CatDetalles getDetalleId() {  
 return detalleId;  
 }  
  
 public void setDetalleId(CatDetalles detalleId) {  
 this.detalleId = detalleId;  
 }  
  
 @XmlTransient  
 public Set<CatOpcionTraduccion> getCatOpcionTraduccionSet() {  
 return catOpcionTraduccionSet;  
 }  
  
 public void setCatOpcionTraduccionSet(Set<CatOpcionTraduccion> catOpcionTraduccionSet) {  
 this.catOpcionTraduccionSet = catOpcionTraduccionSet;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (detalleOpcionesId != null ? detalleOpcionesId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatDetalleOpciones)) {  
 return false;  
 }  
 CatDetalleOpciones other = (CatDetalleOpciones) object;  
 if ((this.detalleOpcionesId == null && other.detalleOpcionesId != null) || (this.detalleOpcionesId != null && !this.detalleOpcionesId.equals(other.detalleOpcionesId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatDetalleOpciones[ detalleOpcionesId=" + detalleOpcionesId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatDetalles.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import java.util.Set;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlTransient;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_DETALLES")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatDetalles.findAll", query = "SELECT c FROM CatDetalles c")  
 , @NamedQuery(name = "CatDetalles.findByDetalleId", query = "SELECT c FROM CatDetalles c WHERE c.detalleId = :detalleId")  
 , @NamedQuery(name = "CatDetalles.findByDetalle", query = "SELECT c FROM CatDetalles c WHERE c.detalle = :detalle")  
 , @NamedQuery(name = "CatDetalles.findByDescripcion", query = "SELECT c FROM CatDetalles c WHERE c.descripcion = :descripcion")})  
public class CatDetalles implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "DETALLE\_ID")  
 private BigDecimal detalleId;  
 @Size(max = 1040)  
 @Column(name = "DETALLE")  
 private String detalle;  
 @Size(max = 1040)  
 @Column(name = "DESCRIPCION")  
 private String descripcion;  
 @OneToMany(mappedBy = "detalleId")  
 private Set<CatConfiguracionDetalle> catConfiguracionDetalleSet;  
 @OneToMany(mappedBy = "detalleId")  
 private Set<CatDetalleOpciones> catDetalleOpcionesSet;  
  
 public CatDetalles() {  
 }  
  
 public CatDetalles(BigDecimal detalleId) {  
 this.detalleId = detalleId;  
 }  
  
 public BigDecimal getDetalleId() {  
 return detalleId;  
 }  
  
 public void setDetalleId(BigDecimal detalleId) {  
 this.detalleId = detalleId;  
 }  
  
 public String getDetalle() {  
 return detalle;  
 }  
  
 public void setDetalle(String detalle) {  
 this.detalle = detalle;  
 }  
  
 public String getDescripcion() {  
 return descripcion;  
 }  
  
 public void setDescripcion(String descripcion) {  
 this.descripcion = descripcion;  
 }  
  
 @XmlTransient  
 public Set<CatConfiguracionDetalle> getCatConfiguracionDetalleSet() {  
 return catConfiguracionDetalleSet;  
 }  
  
 public void setCatConfiguracionDetalleSet(Set<CatConfiguracionDetalle> catConfiguracionDetalleSet) {  
 this.catConfiguracionDetalleSet = catConfiguracionDetalleSet;  
 }  
  
 @XmlTransient  
 public Set<CatDetalleOpciones> getCatDetalleOpcionesSet() {  
 return catDetalleOpcionesSet;  
 }  
  
 public void setCatDetalleOpcionesSet(Set<CatDetalleOpciones> catDetalleOpcionesSet) {  
 this.catDetalleOpcionesSet = catDetalleOpcionesSet;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (detalleId != null ? detalleId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatDetalles)) {  
 return false;  
 }  
 CatDetalles other = (CatDetalles) object;  
 if ((this.detalleId == null && other.detalleId != null) || (this.detalleId != null && !this.detalleId.equals(other.detalleId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatDetalles[ detalleId=" + detalleId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatEstados.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatEstados generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_ESTADOS"  
)  
public class CatEstados implements java.io.Serializable {  
  
  
 private BigDecimal estadoId;  
 private CatPaises catPaises;  
 private String nombre;  
 private Boolean activo;  
 private String claveSat;  
 private Set<CatDatosFiscales> catDatosFiscaleses = new HashSet<CatDatosFiscales>(0);  
  
 public CatEstados() {  
 }  
  
 public CatEstados(CatPaises catPaises, String nombre, Boolean activo, String claveSat, Set<CatDatosFiscales> catDatosFiscaleses) {  
 this.catPaises = catPaises;  
 this.nombre = nombre;  
 this.activo = activo;  
 this.claveSat = claveSat;  
 this.catDatosFiscaleses = catDatosFiscaleses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="ESTADO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getEstadoId() {  
 return this.estadoId;  
 }  
   
 public void setEstadoId(BigDecimal estadoId) {  
 this.estadoId = estadoId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="PAIS\_ID")  
 public CatPaises getCatPaises() {  
 return this.catPaises;  
 }  
   
 public void setCatPaises(CatPaises catPaises) {  
 this.catPaises = catPaises;  
 }  
  
   
 @Column(name="NOMBRE")  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="ACTIVO", precision=1, scale=0)  
 public Boolean getActivo() {  
 return this.activo;  
 }  
   
 public void setActivo(Boolean activo) {  
 this.activo = activo;  
 }  
  
   
 @Column(name="CLAVE\_SAT", length=10)  
 public String getClaveSat() {  
 return this.claveSat;  
 }  
   
 public void setClaveSat(String claveSat) {  
 this.claveSat = claveSat;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catEstados")  
 public Set<CatDatosFiscales> getCatDatosFiscaleses() {  
 return this.catDatosFiscaleses;  
 }  
   
 public void setCatDatosFiscaleses(Set<CatDatosFiscales> catDatosFiscaleses) {  
 this.catDatosFiscaleses = catDatosFiscaleses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatEstatusBitacora.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
  
import java.math.BigDecimal;  
  
import java.util.Set;  
  
import javax.persistence.Basic;  
import javax.persistence.CascadeType;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlTransient;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_ESTATUS\_BITACORAS")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatEstatusBitacora.findAll", query = "SELECT c FROM CatEstatusBitacora c")  
 , @NamedQuery(name = "CatEstatusBitacora.findByEstatusBitacoraId", query = "SELECT c FROM CatEstatusBitacora c WHERE c.estatusBitacoraId = :estatusBitacoraId")  
 , @NamedQuery(name = "CatEstatusBitacora.findByEstatus", query = "SELECT c FROM CatEstatusBitacora c WHERE c.estatus = :estatus")})  
public class CatEstatusBitacora implements Serializable {  
  
 @OneToMany(cascade = CascadeType.ALL, mappedBy = "estatusBitacoraId")  
 private Set<TxBitacora> txBitacoraSet;  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "ESTATUS\_BITACORA\_ID")  
 private BigDecimal estatusBitacoraId;  
 @Basic(optional = false)  
 @NotNull  
 @Size(min = 1, max = 200)  
 @Column(name = "ESTATUS")  
 private String estatus;  
  
 public CatEstatusBitacora() {  
 }  
  
 public CatEstatusBitacora(BigDecimal estatusBitacoraId) {  
 this.estatusBitacoraId = estatusBitacoraId;  
 }  
  
 public CatEstatusBitacora(BigDecimal estatusBitacoraId, String estatus) {  
 this.estatusBitacoraId = estatusBitacoraId;  
 this.estatus = estatus;  
 }  
  
 public BigDecimal getEstatusBitacoraId() {  
 return estatusBitacoraId;  
 }  
  
 public void setEstatusBitacoraId(BigDecimal estatusBitacoraId) {  
 this.estatusBitacoraId = estatusBitacoraId;  
 }  
  
 public String getEstatus() {  
 return estatus;  
 }  
  
 public void setEstatus(String estatus) {  
 this.estatus = estatus;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (estatusBitacoraId != null ? estatusBitacoraId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatEstatusBitacora)) {  
 return false;  
 }  
 CatEstatusBitacora other = (CatEstatusBitacora) object;  
 if ((this.estatusBitacoraId == null && other.estatusBitacoraId != null) || (this.estatusBitacoraId != null && !this.estatusBitacoraId.equals(other.estatusBitacoraId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatEstatusBitacora[ estatusBitacoraId=" + estatusBitacoraId + " ]";  
 }  
  
 @XmlTransient  
 public Set<TxBitacora> getTxBitacoraSet() {  
 return txBitacoraSet;  
 }  
  
 public void setTxBitacoraSet(Set<TxBitacora> txBitacoraSet) {  
 this.txBitacoraSet = txBitacoraSet;  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatEstatusFactura.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatEstatusFactura generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_ESTATUS\_FACTURA"  
)  
public class CatEstatusFactura implements java.io.Serializable {  
  
  
 private BigDecimal estatusFacturaId;  
 private String nombre;  
 private Set<TxDocumentos> txDocumentoses = new HashSet<TxDocumentos>(0);  
  
 public CatEstatusFactura() {  
 }  
  
 public CatEstatusFactura(String nombre, Set<TxDocumentos> txDocumentoses) {  
 this.nombre = nombre;  
 this.txDocumentoses = txDocumentoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="ESTATUS\_FACTURA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getEstatusFacturaId() {  
 return this.estatusFacturaId;  
 }  
   
 public void setEstatusFacturaId(BigDecimal estatusFacturaId) {  
 this.estatusFacturaId = estatusFacturaId;  
 }  
  
   
 @Column(name="NOMBRE", length=50)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catEstatusFactura")  
 public Set<TxDocumentos> getTxDocumentoses() {  
 return this.txDocumentoses;  
 }  
   
 public void setTxDocumentoses(Set<TxDocumentos> txDocumentoses) {  
 this.txDocumentoses = txDocumentoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatEstatusPagos.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatEstatusPagos generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_ESTATUS\_PAGOS"  
)  
public class CatEstatusPagos implements java.io.Serializable {  
  
  
 private BigDecimal estatusPagoId;  
 private String nombre;  
 private Set<TxDocumentos> txDocumentoses = new HashSet<TxDocumentos>(0);  
  
 public CatEstatusPagos() {  
 }  
  
 public CatEstatusPagos(String nombre, Set<TxDocumentos> txDocumentoses) {  
 this.nombre = nombre;  
 this.txDocumentoses = txDocumentoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="ESTATUS\_PAGO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getEstatusPagoId() {  
 return this.estatusPagoId;  
 }  
   
 public void setEstatusPagoId(BigDecimal estatusPagoId) {  
 this.estatusPagoId = estatusPagoId;  
 }  
  
   
 @Column(name="NOMBRE", length=100)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catEstatusPagos")  
 public Set<TxDocumentos> getTxDocumentoses() {  
 return this.txDocumentoses;  
 }  
   
 public void setTxDocumentoses(Set<TxDocumentos> txDocumentoses) {  
 this.txDocumentoses = txDocumentoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatEstatusPerfil.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import java.util.HashSet;  
import java.util.Set;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatEstatusPerfil generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_ESTATUS\_PERFIL"  
)  
public class CatEstatusPerfil implements java.io.Serializable {  
  
  
 private BigDecimal estatusPerfilId;  
 private String estatusPerfil;  
 private String color;  
 private Set<TxPerfilesVariable> txPerfilesVariables = new HashSet<TxPerfilesVariable>(0);  
  
 public CatEstatusPerfil() {  
 }  
  
 public CatEstatusPerfil(String estatusPerfil, Set<TxPerfilesVariable> txPerfilesVariables, String color) {  
 this.estatusPerfil = estatusPerfil;  
 this.txPerfilesVariables = txPerfilesVariables;  
 this.color = color;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="ESTATUS\_PERFIL\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getEstatusPerfilId() {  
 return this.estatusPerfilId;  
 }  
   
 public void setEstatusPerfilId(BigDecimal estatusPerfilId) {  
 this.estatusPerfilId = estatusPerfilId;  
 }  
  
   
 @Column(name="ESTATUS\_PERFIL", length=150)  
 public String getEstatusPerfil() {  
 return this.estatusPerfil;  
 }  
   
 public void setEstatusPerfil(String estatusPerfil) {  
 this.estatusPerfil = estatusPerfil;  
 }  
   
 @Column(name="COLOR", length=20)  
 public String getColor() {  
 return this.color;  
 }  
   
 public void setColor(String color) {  
 this.color = color;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catEstatusPerfil")  
 public Set<TxPerfilesVariable> getTxPerfilesVariables() {  
 return this.txPerfilesVariables;  
 }  
   
 public void setTxPerfilesVariables(Set<TxPerfilesVariable> txPerfilesVariables) {  
 this.txPerfilesVariables = txPerfilesVariables;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatEstatusVariable.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatEstatusVariable generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_ESTATUS\_VARIABLE"  
)  
public class CatEstatusVariable implements java.io.Serializable {  
  
  
 private BigDecimal estatusVariableId;  
 private BigDecimal estatusVarId;  
 private String nombre;  
 private String color;  
  
 public CatEstatusVariable() {  
 }  
  
 public CatEstatusVariable(BigDecimal estatusVarId, String nombre, String color) {  
 this.estatusVarId = estatusVarId;  
 this.nombre = nombre;  
 this.color = color;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="ESTATUS\_VARIABLE\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getEstatusVariableId() {  
 return this.estatusVariableId;  
 }  
   
 public void setEstatusVariableId(BigDecimal estatusVariableId) {  
 this.estatusVariableId = estatusVariableId;  
 }  
  
   
 @Column(name="ESTATUS\_VAR\_ID", precision=22, scale=0)  
 public BigDecimal getEstatusVarId() {  
 return this.estatusVarId;  
 }  
   
 public void setEstatusVarId(BigDecimal estatusVarId) {  
 this.estatusVarId = estatusVarId;  
 }  
  
   
 @Column(name="NOMBRE", length=150)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
   
 @Column(name="COLOR", length=20)  
 public String getColor() {  
 return this.color;  
 }  
   
 public void setColor(String color) {  
 this.color = color;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatEtapas.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatEtapas generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_ETAPAS"  
)  
public class CatEtapas implements java.io.Serializable {  
  
  
 private BigDecimal etapaId;  
 private String etapa;  
 private Set<TxEscenario> txEscenarios = new HashSet<TxEscenario>(0);  
  
 public CatEtapas() {  
 }  
  
 public CatEtapas(String etapa, Set<TxEscenario> txEscenarios) {  
 this.etapa = etapa;  
 this.txEscenarios = txEscenarios;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="ETAPA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getEtapaId() {  
 return this.etapaId;  
 }  
   
 public void setEtapaId(BigDecimal etapaId) {  
 this.etapaId = etapaId;  
 }  
  
   
 @Column(name="ETAPA", length=200)  
 public String getEtapa() {  
 return this.etapa;  
 }  
   
 public void setEtapa(String etapa) {  
 this.etapa = etapa;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catEtapas")  
 public Set<TxEscenario> getTxEscenarios() {  
 return this.txEscenarios;  
 }  
   
 public void setTxEscenarios(Set<TxEscenario> txEscenarios) {  
 this.txEscenarios = txEscenarios;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatEventos.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import java.math.BigInteger;  
import java.util.Date;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
import javax.xml.bind.annotation.XmlRootElement;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_EVENTOS")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatEventos.findAll", query = "SELECT c FROM CatEventos c")  
 , @NamedQuery(name = "CatEventos.findByEventoId", query = "SELECT c FROM CatEventos c WHERE c.eventoId = :eventoId")  
 , @NamedQuery(name = "CatEventos.findByNombre", query = "SELECT c FROM CatEventos c WHERE c.nombre = :nombre")  
 , @NamedQuery(name = "CatEventos.findByDescripcion", query = "SELECT c FROM CatEventos c WHERE c.descripcion = :descripcion")  
 , @NamedQuery(name = "CatEventos.findByRequierePotencia", query = "SELECT c FROM CatEventos c WHERE c.requierePotencia = :requierePotencia")  
 , @NamedQuery(name = "CatEventos.findByRequiereOt", query = "SELECT c FROM CatEventos c WHERE c.requiereOt = :requiereOt")  
 , @NamedQuery(name = "CatEventos.findByEsFalla", query = "SELECT c FROM CatEventos c WHERE c.esFalla = :esFalla")  
 , @NamedQuery(name = "CatEventos.findByMemRequiereId", query = "SELECT c FROM CatEventos c WHERE c.memRequiereId = :memRequiereId")  
 , @NamedQuery(name = "CatEventos.findByRequiereMwOfertados", query = "SELECT c FROM CatEventos c WHERE c.requiereMwOfertados = :requiereMwOfertados")  
 , @NamedQuery(name = "CatEventos.findByRequiereLicencia", query = "SELECT c FROM CatEventos c WHERE c.requiereLicencia = :requiereLicencia")  
 , @NamedQuery(name = "CatEventos.findByRequiereOperadorCenace", query = "SELECT c FROM CatEventos c WHERE c.requiereOperadorCenace = :requiereOperadorCenace")  
 , @NamedQuery(name = "CatEventos.findByRequiereConcepto", query = "SELECT c FROM CatEventos c WHERE c.requiereConcepto = :requiereConcepto")  
 , @NamedQuery(name = "CatEventos.findByActivo", query = "SELECT c FROM CatEventos c WHERE c.activo = :activo")  
 , @NamedQuery(name = "CatEventos.findByObservaciones", query = "SELECT c FROM CatEventos c WHERE c.observaciones = :observaciones")  
 , @NamedQuery(name = "CatEventos.findByFechaRegistro", query = "SELECT c FROM CatEventos c WHERE c.fechaRegistro = :fechaRegistro")  
 , @NamedQuery(name = "CatEventos.findByUsuarioRegistro", query = "SELECT c FROM CatEventos c WHERE c.usuarioRegistro = :usuarioRegistro")  
 , @NamedQuery(name = "CatEventos.findByFechaModificacion", query = "SELECT c FROM CatEventos c WHERE c.fechaModificacion = :fechaModificacion")  
 , @NamedQuery(name = "CatEventos.findByUsuarioModifico", query = "SELECT c FROM CatEventos c WHERE c.usuarioModifico = :usuarioModifico")})  
public class CatEventos implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "EVENTO\_ID")  
 private BigDecimal eventoId;  
 @Size(max = 1020)  
 @Column(name = "NOMBRE")  
 private String nombre;  
 @Column(name = "DESCRIPCION")  
 private BigInteger descripcion;  
 @Column(name = "REQUIERE\_POTENCIA")  
 private BigInteger requierePotencia;  
 @Column(name = "REQUIERE\_OT")  
 private BigInteger requiereOt;  
 @Column(name = "ES\_FALLA")  
 private BigInteger esFalla;  
 @Column(name = "MEM\_REQUIERE\_ID")  
 private BigInteger memRequiereId;  
 @Column(name = "REQUIERE\_MW\_OFERTADOS")  
 private BigInteger requiereMwOfertados;  
 @Column(name = "REQUIERE\_LICENCIA")  
 private BigInteger requiereLicencia;  
 @Column(name = "REQUIERE\_OPERADOR\_CENACE")  
 private BigInteger requiereOperadorCenace;  
 @Column(name = "REQUIERE\_CONCEPTO")  
 private BigInteger requiereConcepto;  
 @Column(name = "ACTIVO")  
 private BigInteger activo;  
 @Size(max = 4000)  
 @Column(name = "OBSERVACIONES")  
 private String observaciones;  
 @Column(name = "FECHA\_REGISTRO")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaRegistro;  
 @Column(name = "USUARIO\_REGISTRO")  
 private BigInteger usuarioRegistro;  
 @Column(name = "FECHA\_MODIFICACION")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaModificacion;  
 @Column(name = "USUARIO\_MODIFICO")  
 private BigInteger usuarioModifico;  
 @JoinColumn(name = "CLASIFICACION\_EVENTO\_ID", referencedColumnName = "CLASIFICACION\_EVENTO\_ID")  
 @ManyToOne  
 private CatClasificacionEventos clasificacionEventoId;  
  
 public CatEventos() {  
 }  
  
 public CatEventos(BigDecimal eventoId) {  
 this.eventoId = eventoId;  
 }  
  
 public BigDecimal getEventoId() {  
 return eventoId;  
 }  
  
 public void setEventoId(BigDecimal eventoId) {  
 this.eventoId = eventoId;  
 }  
  
 public String getNombre() {  
 return nombre;  
 }  
  
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
 public BigInteger getDescripcion() {  
 return descripcion;  
 }  
  
 public void setDescripcion(BigInteger descripcion) {  
 this.descripcion = descripcion;  
 }  
  
 public BigInteger getRequierePotencia() {  
 return requierePotencia;  
 }  
  
 public void setRequierePotencia(BigInteger requierePotencia) {  
 this.requierePotencia = requierePotencia;  
 }  
  
 public BigInteger getRequiereOt() {  
 return requiereOt;  
 }  
  
 public void setRequiereOt(BigInteger requiereOt) {  
 this.requiereOt = requiereOt;  
 }  
  
 public BigInteger getEsFalla() {  
 return esFalla;  
 }  
  
 public void setEsFalla(BigInteger esFalla) {  
 this.esFalla = esFalla;  
 }  
  
 public BigInteger getMemRequiereId() {  
 return memRequiereId;  
 }  
  
 public void setMemRequiereId(BigInteger memRequiereId) {  
 this.memRequiereId = memRequiereId;  
 }  
  
 public BigInteger getRequiereMwOfertados() {  
 return requiereMwOfertados;  
 }  
  
 public void setRequiereMwOfertados(BigInteger requiereMwOfertados) {  
 this.requiereMwOfertados = requiereMwOfertados;  
 }  
  
 public BigInteger getRequiereLicencia() {  
 return requiereLicencia;  
 }  
  
 public void setRequiereLicencia(BigInteger requiereLicencia) {  
 this.requiereLicencia = requiereLicencia;  
 }  
  
 public BigInteger getRequiereOperadorCenace() {  
 return requiereOperadorCenace;  
 }  
  
 public void setRequiereOperadorCenace(BigInteger requiereOperadorCenace) {  
 this.requiereOperadorCenace = requiereOperadorCenace;  
 }  
  
 public BigInteger getRequiereConcepto() {  
 return requiereConcepto;  
 }  
  
 public void setRequiereConcepto(BigInteger requiereConcepto) {  
 this.requiereConcepto = requiereConcepto;  
 }  
  
 public BigInteger getActivo() {  
 return activo;  
 }  
  
 public void setActivo(BigInteger activo) {  
 this.activo = activo;  
 }  
  
 public String getObservaciones() {  
 return observaciones;  
 }  
  
 public void setObservaciones(String observaciones) {  
 this.observaciones = observaciones;  
 }  
  
 public Date getFechaRegistro() {  
 return fechaRegistro;  
 }  
  
 public void setFechaRegistro(Date fechaRegistro) {  
 this.fechaRegistro = fechaRegistro;  
 }  
  
 public BigInteger getUsuarioRegistro() {  
 return usuarioRegistro;  
 }  
  
 public void setUsuarioRegistro(BigInteger usuarioRegistro) {  
 this.usuarioRegistro = usuarioRegistro;  
 }  
  
 public Date getFechaModificacion() {  
 return fechaModificacion;  
 }  
  
 public void setFechaModificacion(Date fechaModificacion) {  
 this.fechaModificacion = fechaModificacion;  
 }  
  
 public BigInteger getUsuarioModifico() {  
 return usuarioModifico;  
 }  
  
 public void setUsuarioModifico(BigInteger usuarioModifico) {  
 this.usuarioModifico = usuarioModifico;  
 }  
  
 public CatClasificacionEventos getClasificacionEventoId() {  
 return clasificacionEventoId;  
 }  
  
 public void setClasificacionEventoId(CatClasificacionEventos clasificacionEventoId) {  
 this.clasificacionEventoId = clasificacionEventoId;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (eventoId != null ? eventoId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatEventos)) {  
 return false;  
 }  
 CatEventos other = (CatEventos) object;  
 if ((this.eventoId == null && other.eventoId != null) || (this.eventoId != null && !this.eventoId.equals(other.eventoId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "com.test.entity.CatEventos[ eventoId=" + eventoId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatFactorTipo.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatFactorTipo generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_FACTOR\_TIPO"  
)  
public class CatFactorTipo implements java.io.Serializable {  
  
  
 private BigDecimal factorTipoId;  
 private String nombre;  
 private Set<CatCurvasCorreccion> catCurvasCorreccions = new HashSet<CatCurvasCorreccion>(0);  
 private Set<CatLimitesDominio> catLimitesDominios = new HashSet<CatLimitesDominio>(0);  
  
 public CatFactorTipo() {  
 }  
  
 public CatFactorTipo(String nombre, Set<CatCurvasCorreccion> catCurvasCorreccions, Set<CatLimitesDominio> catLimitesDominios) {  
 this.nombre = nombre;  
 this.catCurvasCorreccions = catCurvasCorreccions;  
 this.catLimitesDominios = catLimitesDominios;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="FACTOR\_TIPO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getFactorTipoId() {  
 return this.factorTipoId;  
 }  
   
 public void setFactorTipoId(BigDecimal factorTipoId) {  
 this.factorTipoId = factorTipoId;  
 }  
  
   
 @Column(name="NOMBRE", length=100)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catFactorTipo")  
 public Set<CatCurvasCorreccion> getCatCurvasCorreccions() {  
 return this.catCurvasCorreccions;  
 }  
   
 public void setCatCurvasCorreccions(Set<CatCurvasCorreccion> catCurvasCorreccions) {  
 this.catCurvasCorreccions = catCurvasCorreccions;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catFactorTipo")  
 public Set<CatLimitesDominio> getCatLimitesDominios() {  
 return this.catLimitesDominios;  
 }  
   
 public void setCatLimitesDominios(Set<CatLimitesDominio> catLimitesDominios) {  
 this.catLimitesDominios = catLimitesDominios;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatFacturaProductos.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatFacturaProductos generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_FACTURA\_PRODUCTOS"  
)  
public class CatFacturaProductos implements java.io.Serializable {  
  
  
 private BigDecimal facturaProductoId;  
 private CatServicios catServicios;  
 private TxDocumentos txDocumentos;  
 private BigDecimal cantidad;  
 private BigDecimal precioUnitario;  
 private BigDecimal descuento;  
 private BigDecimal importe;  
 private String observaciones;  
  
 public CatFacturaProductos() {  
 }  
  
 public CatFacturaProductos(CatServicios catServicios, TxDocumentos txDocumentos, BigDecimal cantidad, BigDecimal precioUnitario, BigDecimal descuento, BigDecimal importe, String observaciones) {  
 this.catServicios = catServicios;  
 this.txDocumentos = txDocumentos;  
 this.cantidad = cantidad;  
 this.precioUnitario = precioUnitario;  
 this.descuento = descuento;  
 this.importe = importe;  
 this.observaciones = observaciones;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="FACTURA\_PRODUCTO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getFacturaProductoId() {  
 return this.facturaProductoId;  
 }  
   
 public void setFacturaProductoId(BigDecimal facturaProductoId) {  
 this.facturaProductoId = facturaProductoId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="SERVICIO\_ID")  
 public CatServicios getCatServicios() {  
 return this.catServicios;  
 }  
   
 public void setCatServicios(CatServicios catServicios) {  
 this.catServicios = catServicios;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="DOCUMENTO\_ID")  
 public TxDocumentos getTxDocumentos() {  
 return this.txDocumentos;  
 }  
   
 public void setTxDocumentos(TxDocumentos txDocumentos) {  
 this.txDocumentos = txDocumentos;  
 }  
  
   
 @Column(name="CANTIDAD", precision=22, scale=0)  
 public BigDecimal getCantidad() {  
 return this.cantidad;  
 }  
   
 public void setCantidad(BigDecimal cantidad) {  
 this.cantidad = cantidad;  
 }  
  
   
 @Column(name="PRECIO\_UNITARIO", precision=22, scale=0)  
 public BigDecimal getPrecioUnitario() {  
 return this.precioUnitario;  
 }  
   
 public void setPrecioUnitario(BigDecimal precioUnitario) {  
 this.precioUnitario = precioUnitario;  
 }  
  
   
 @Column(name="DESCUENTO", precision=22, scale=0)  
 public BigDecimal getDescuento() {  
 return this.descuento;  
 }  
   
 public void setDescuento(BigDecimal descuento) {  
 this.descuento = descuento;  
 }  
  
   
 @Column(name="IMPORTE", precision=22, scale=0)  
 public BigDecimal getImporte() {  
 return this.importe;  
 }  
   
 public void setImporte(BigDecimal importe) {  
 this.importe = importe;  
 }  
  
   
 @Column(name="OBSERVACIONES")  
 public String getObservaciones() {  
 return this.observaciones;  
 }  
   
 public void setObservaciones(String observaciones) {  
 this.observaciones = observaciones;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatFirmaCincominutal.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatFirmaCincominutal generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_FIRMA\_CINCOMINUTAL"  
)  
public class CatFirmaCincominutal implements java.io.Serializable {  
  
  
 private BigDecimal firmaId;  
 private String firma;  
 private Set<CatMesDiaFirma> catMesDiaFirmas = new HashSet<CatMesDiaFirma>(0);  
  
 public CatFirmaCincominutal() {  
 }  
  
 public CatFirmaCincominutal(String firma, Set<CatMesDiaFirma> catMesDiaFirmas) {  
 this.firma = firma;  
 this.catMesDiaFirmas = catMesDiaFirmas;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="FIRMA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getFirmaId() {  
 return this.firmaId;  
 }  
   
 public void setFirmaId(BigDecimal firmaId) {  
 this.firmaId = firmaId;  
 }  
  
   
 @Column(name="FIRMA", length=200)  
 public String getFirma() {  
 return this.firma;  
 }  
   
 public void setFirma(String firma) {  
 this.firma = firma;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catFirmaCincominutal")  
 public Set<CatMesDiaFirma> getCatMesDiaFirmas() {  
 return this.catMesDiaFirmas;  
 }  
   
 public void setCatMesDiaFirmas(Set<CatMesDiaFirma> catMesDiaFirmas) {  
 this.catMesDiaFirmas = catMesDiaFirmas;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatFormasPago.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatFormasPago generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_FORMAS\_PAGO"  
)  
public class CatFormasPago implements java.io.Serializable {  
  
  
 private BigDecimal formaPagoId;  
 private String nombre;  
 private Set<TxDocumentos> txDocumentoses = new HashSet<TxDocumentos>(0);  
  
 public CatFormasPago() {  
 }  
  
 public CatFormasPago(String nombre, Set<TxDocumentos> txDocumentoses) {  
 this.nombre = nombre;  
 this.txDocumentoses = txDocumentoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="FORMA\_PAGO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getFormaPagoId() {  
 return this.formaPagoId;  
 }  
   
 public void setFormaPagoId(BigDecimal formaPagoId) {  
 this.formaPagoId = formaPagoId;  
 }  
  
   
 @Column(name="NOMBRE")  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catFormasPago")  
 public Set<TxDocumentos> getTxDocumentoses() {  
 return this.txDocumentoses;  
 }  
   
 public void setTxDocumentoses(Set<TxDocumentos> txDocumentoses) {  
 this.txDocumentoses = txDocumentoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatFormulaComponentes.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import java.util.HashSet;  
import java.util.Set;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatFormulaComponentes generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_FORMULA\_COMPONENTES"  
)  
public class CatFormulaComponentes implements java.io.Serializable {  
  
  
 private BigDecimal catFormulaComponentesId;  
 private TxFormulas txFormulas;  
 private CatComponentes catComponentes;  
 private BigDecimal valor;  
 private String instanciaEstampaTiempo;  
 private Set<CatFormulaComponentesTxEvento> catFormulaComponentesTxEventos = new HashSet<CatFormulaComponentesTxEvento>(0);  
  
 public CatFormulaComponentes() {  
 }  
  
 public CatFormulaComponentes(TxFormulas txFormulas, CatComponentes catComponentes, BigDecimal valor, String instanciaEstampaTiempo, Set<CatFormulaComponentesTxEvento> catFormulaComponentesTxEventos) {  
 this.txFormulas = txFormulas;  
 this.catComponentes = catComponentes;  
 this.valor = valor;  
 this.instanciaEstampaTiempo = instanciaEstampaTiempo;  
 this.catFormulaComponentesTxEventos = catFormulaComponentesTxEventos;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="CAT\_FORMULA\_COMPONENTES\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getCatFormulaComponentesId() {  
 return this.catFormulaComponentesId;  
 }  
   
 public void setCatFormulaComponentesId(BigDecimal catFormulaComponentesId) {  
 this.catFormulaComponentesId = catFormulaComponentesId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="FORMULA\_ID")  
 public TxFormulas getTxFormulas() {  
 return this.txFormulas;  
 }  
   
 public void setTxFormulas(TxFormulas txFormulas) {  
 this.txFormulas = txFormulas;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="COMPONENTE\_ID")  
 public CatComponentes getCatComponentes() {  
 return this.catComponentes;  
 }  
   
 public void setCatComponentes(CatComponentes catComponentes) {  
 this.catComponentes = catComponentes;  
 }  
  
   
 @Column(name="VALOR", precision=22, scale=0)  
 public BigDecimal getValor() {  
 return this.valor;  
 }  
   
 public void setValor(BigDecimal valor) {  
 this.valor = valor;  
 }  
  
   
 @Column(name="INSTANCIA\_ESTAMPA\_TIEMPO")  
 public String getInstanciaEstampaTiempo() {  
 return this.instanciaEstampaTiempo;  
 }  
   
 public void setInstanciaEstampaTiempo(String instanciaEstampaTiempo) {  
 this.instanciaEstampaTiempo = instanciaEstampaTiempo;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catFormulaComponentes")  
 public Set<CatFormulaComponentesTxEvento> getCatFormulaComponentesTxEventos() {  
 return this.catFormulaComponentesTxEventos;  
 }  
   
 public void setCatFormulaComponentesTxEventos(Set<CatFormulaComponentesTxEvento> catFormulaComponentesTxEventos) {  
 this.catFormulaComponentesTxEventos = catFormulaComponentesTxEventos;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatFormulaComponentesTxEvento.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatFormulaComponentesTxEvento generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_FORMULA\_COMPONENTES\_TX\_EVENTO"  
)  
public class CatFormulaComponentesTxEvento implements java.io.Serializable {  
  
  
 private BigDecimal catFormulaComponentesTxEventoId;  
 private TxEvento txEvento;  
 private CatFormulaComponentes catFormulaComponentes;  
  
 public CatFormulaComponentesTxEvento() {  
 }  
  
 public CatFormulaComponentesTxEvento(TxEvento txEvento, CatFormulaComponentes catFormulaComponentes) {  
 this.txEvento = txEvento;  
 this.catFormulaComponentes = catFormulaComponentes;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="CAT\_FORMULA\_COMPONENTES\_TX\_EVENTO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getCatFormulaComponentesTxEventoId() {  
 return this.catFormulaComponentesTxEventoId;  
 }  
   
 public void setCatFormulaComponentesTxEventoId(BigDecimal catFormulaComponentesTxEventoId) {  
 this.catFormulaComponentesTxEventoId = catFormulaComponentesTxEventoId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="EVENTO\_ID")  
 public TxEvento getTxEvento() {  
 return this.txEvento;  
 }  
   
 public void setTxEvento(TxEvento txEvento) {  
 this.txEvento = txEvento;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="CAT\_FORMULA\_COMPONENTES\_ID")  
 public CatFormulaComponentes getCatFormulaComponentes() {  
 return this.catFormulaComponentes;  
 }  
   
 public void setCatFormulaComponentes(CatFormulaComponentes catFormulaComponentes) {  
 this.catFormulaComponentes = catFormulaComponentes;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatImpuestos.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatImpuestos generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_IMPUESTOS"  
)  
public class CatImpuestos implements java.io.Serializable {  
  
  
 private BigDecimal impuestoId;  
 private CatTipoImpuestos catTipoImpuestos;  
 private String nombre;  
 private BigDecimal porcentaje;  
 private Boolean exento;  
 private String descripcion;  
 private Boolean activo;  
 private Set<CatServicioImpuestos> catServicioImpuestoses = new HashSet<CatServicioImpuestos>(0);  
  
 public CatImpuestos() {  
 }  
  
 public CatImpuestos(CatTipoImpuestos catTipoImpuestos, String nombre, BigDecimal porcentaje, Boolean exento, String descripcion, Boolean activo, Set<CatServicioImpuestos> catServicioImpuestoses) {  
 this.catTipoImpuestos = catTipoImpuestos;  
 this.nombre = nombre;  
 this.porcentaje = porcentaje;  
 this.exento = exento;  
 this.descripcion = descripcion;  
 this.activo = activo;  
 this.catServicioImpuestoses = catServicioImpuestoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="IMPUESTO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getImpuestoId() {  
 return this.impuestoId;  
 }  
   
 public void setImpuestoId(BigDecimal impuestoId) {  
 this.impuestoId = impuestoId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="TIPO\_IMPUESTO\_ID")  
 public CatTipoImpuestos getCatTipoImpuestos() {  
 return this.catTipoImpuestos;  
 }  
   
 public void setCatTipoImpuestos(CatTipoImpuestos catTipoImpuestos) {  
 this.catTipoImpuestos = catTipoImpuestos;  
 }  
  
   
 @Column(name="NOMBRE")  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="PORCENTAJE", precision=22, scale=0)  
 public BigDecimal getPorcentaje() {  
 return this.porcentaje;  
 }  
   
 public void setPorcentaje(BigDecimal porcentaje) {  
 this.porcentaje = porcentaje;  
 }  
  
   
 @Column(name="EXENTO", precision=1, scale=0)  
 public Boolean getExento() {  
 return this.exento;  
 }  
   
 public void setExento(Boolean exento) {  
 this.exento = exento;  
 }  
  
   
 @Column(name="DESCRIPCION")  
 public String getDescripcion() {  
 return this.descripcion;  
 }  
   
 public void setDescripcion(String descripcion) {  
 this.descripcion = descripcion;  
 }  
  
   
 @Column(name="ACTIVO", precision=1, scale=0)  
 public Boolean getActivo() {  
 return this.activo;  
 }  
   
 public void setActivo(Boolean activo) {  
 this.activo = activo;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catImpuestos")  
 public Set<CatServicioImpuestos> getCatServicioImpuestoses() {  
 return this.catServicioImpuestoses;  
 }  
   
 public void setCatServicioImpuestoses(Set<CatServicioImpuestos> catServicioImpuestoses) {  
 this.catServicioImpuestoses = catServicioImpuestoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatLenguajes.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import java.util.Set;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlTransient;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_LENGUAJES")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatLenguajes.findAll", query = "SELECT c FROM CatLenguajes c")  
 , @NamedQuery(name = "CatLenguajes.findByLenguajeId", query = "SELECT c FROM CatLenguajes c WHERE c.lenguajeId = :lenguajeId")  
 , @NamedQuery(name = "CatLenguajes.findByLenguaje", query = "SELECT c FROM CatLenguajes c WHERE c.lenguaje = :lenguaje")})  
public class CatLenguajes implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "LENGUAJE\_ID")  
 private BigDecimal lenguajeId;  
 @Size(max = 150)  
 @Column(name = "LENGUAJE")  
 private String lenguaje;  
 @OneToMany(mappedBy = "lenguajeId")  
 private Set<CatOpcionTraduccion> catOpcionTraduccionSet;  
  
 public CatLenguajes() {  
 }  
  
 public CatLenguajes(BigDecimal lenguajeId) {  
 this.lenguajeId = lenguajeId;  
 }  
  
 public BigDecimal getLenguajeId() {  
 return lenguajeId;  
 }  
  
 public void setLenguajeId(BigDecimal lenguajeId) {  
 this.lenguajeId = lenguajeId;  
 }  
  
 public String getLenguaje() {  
 return lenguaje;  
 }  
  
 public void setLenguaje(String lenguaje) {  
 this.lenguaje = lenguaje;  
 }  
  
 @XmlTransient  
 public Set<CatOpcionTraduccion> getCatOpcionTraduccionSet() {  
 return catOpcionTraduccionSet;  
 }  
  
 public void setCatOpcionTraduccionSet(Set<CatOpcionTraduccion> catOpcionTraduccionSet) {  
 this.catOpcionTraduccionSet = catOpcionTraduccionSet;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (lenguajeId != null ? lenguajeId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatLenguajes)) {  
 return false;  
 }  
 CatLenguajes other = (CatLenguajes) object;  
 if ((this.lenguajeId == null && other.lenguajeId != null) || (this.lenguajeId != null && !this.lenguajeId.equals(other.lenguajeId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatLenguajes[ lenguajeId=" + lenguajeId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatLimitesDominio.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatLimitesDominio generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_LIMITES\_DOMINIO"  
)  
public class CatLimitesDominio implements java.io.Serializable {  
  
  
 private BigDecimal limitesDominioId;  
 private CatCapacidad catCapacidad;  
 private CatCombustible catCombustible;  
 private CatFactorTipo catFactorTipo;  
 private CatPlanta catPlanta;  
 private CatVariable catVariable;  
 private BigDecimal limiteInferior;  
 private BigDecimal limiteSuperior;  
 private BigDecimal rangoInferior;  
 private BigDecimal rangoSuperior;  
  
 public CatLimitesDominio() {  
 }  
  
 public CatLimitesDominio(CatCapacidad catCapacidad, CatCombustible catCombustible, CatFactorTipo catFactorTipo, CatPlanta catPlanta, CatVariable catVariable, BigDecimal limiteInferior, BigDecimal limiteSuperior, BigDecimal rangoInferior, BigDecimal rangoSuperior) {  
 this.catCapacidad = catCapacidad;  
 this.catCombustible = catCombustible;  
 this.catFactorTipo = catFactorTipo;  
 this.catPlanta = catPlanta;  
 this.catVariable = catVariable;  
 this.limiteInferior = limiteInferior;  
 this.limiteSuperior = limiteSuperior;  
 this.rangoInferior = rangoInferior;  
 this.rangoSuperior = rangoSuperior;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="LIMITES\_DOMINIO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getLimitesDominioId() {  
 return this.limitesDominioId;  
 }  
   
 public void setLimitesDominioId(BigDecimal limitesDominioId) {  
 this.limitesDominioId = limitesDominioId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="CAPACIDAD\_ID")  
 public CatCapacidad getCatCapacidad() {  
 return this.catCapacidad;  
 }  
   
 public void setCatCapacidad(CatCapacidad catCapacidad) {  
 this.catCapacidad = catCapacidad;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="COMBUSTIBLE\_ID")  
 public CatCombustible getCatCombustible() {  
 return this.catCombustible;  
 }  
   
 public void setCatCombustible(CatCombustible catCombustible) {  
 this.catCombustible = catCombustible;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="FACTOR\_TIPO\_ID")  
 public CatFactorTipo getCatFactorTipo() {  
 return this.catFactorTipo;  
 }  
   
 public void setCatFactorTipo(CatFactorTipo catFactorTipo) {  
 this.catFactorTipo = catFactorTipo;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="PLANTA\_ID")  
 public CatPlanta getCatPlanta() {  
 return this.catPlanta;  
 }  
   
 public void setCatPlanta(CatPlanta catPlanta) {  
 this.catPlanta = catPlanta;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="VARIABLE\_ID")  
 public CatVariable getCatVariable() {  
 return this.catVariable;  
 }  
   
 public void setCatVariable(CatVariable catVariable) {  
 this.catVariable = catVariable;  
 }  
  
   
 @Column(name="LIMITE\_INFERIOR", precision=22, scale=0)  
 public BigDecimal getLimiteInferior() {  
 return this.limiteInferior;  
 }  
   
 public void setLimiteInferior(BigDecimal limiteInferior) {  
 this.limiteInferior = limiteInferior;  
 }  
  
   
 @Column(name="LIMITE\_SUPERIOR", precision=22, scale=0)  
 public BigDecimal getLimiteSuperior() {  
 return this.limiteSuperior;  
 }  
   
 public void setLimiteSuperior(BigDecimal limiteSuperior) {  
 this.limiteSuperior = limiteSuperior;  
 }  
  
   
 @Column(name="RANGO\_INFERIOR", precision=22, scale=0)  
 public BigDecimal getRangoInferior() {  
 return this.rangoInferior;  
 }  
   
 public void setRangoInferior(BigDecimal rangoInferior) {  
 this.rangoInferior = rangoInferior;  
 }  
  
   
 @Column(name="RANGO\_SUPERIOR", precision=22, scale=0)  
 public BigDecimal getRangoSuperior() {  
 return this.rangoSuperior;  
 }  
   
 public void setRangoSuperior(BigDecimal rangoSuperior) {  
 this.rangoSuperior = rangoSuperior;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatMedidor.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatMedidor generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_MEDIDOR"  
)  
public class CatMedidor implements java.io.Serializable {  
  
  
 private BigDecimal medidorId;  
 private String medidor;  
  
 public CatMedidor() {  
 }  
  
 public CatMedidor(String medidor) {  
 this.medidor = medidor;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="MEDIDOR\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getMedidorId() {  
 return this.medidorId;  
 }  
   
 public void setMedidorId(BigDecimal medidorId) {  
 this.medidorId = medidorId;  
 }  
  
   
 @Column(name="MEDIDOR", nullable=false, length=100)  
 public String getMedidor() {  
 return this.medidor;  
 }  
   
 public void setMedidor(String medidor) {  
 this.medidor = medidor;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatMedidores.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatMedidores generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_MEDIDORES"  
)  
public class CatMedidores implements java.io.Serializable {  
  
  
 private BigDecimal medidorId;  
 private CatPlanta catPlanta;  
 private String nombre;  
 private Set<TxPerfilesVariable> txPerfilesVariables = new HashSet<TxPerfilesVariable>(0);  
  
 public CatMedidores() {  
 }  
  
 public CatMedidores(CatPlanta catPlanta, String nombre, Set<TxPerfilesVariable> txPerfilesVariables) {  
 this.catPlanta = catPlanta;  
 this.nombre = nombre;  
 this.txPerfilesVariables = txPerfilesVariables;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="MEDIDOR\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getMedidorId() {  
 return this.medidorId;  
 }  
   
 public void setMedidorId(BigDecimal medidorId) {  
 this.medidorId = medidorId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="PLANTA\_ID")  
 public CatPlanta getCatPlanta() {  
 return this.catPlanta;  
 }  
   
 public void setCatPlanta(CatPlanta catPlanta) {  
 this.catPlanta = catPlanta;  
 }  
  
   
 @Column(name="NOMBRE", length=100)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catMedidores")  
 public Set<TxPerfilesVariable> getTxPerfilesVariables() {  
 return this.txPerfilesVariables;  
 }  
   
 public void setTxPerfilesVariables(Set<TxPerfilesVariable> txPerfilesVariables) {  
 this.txPerfilesVariables = txPerfilesVariables;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatMesCargo.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.Date;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* CatMesCargo generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_MES\_CARGO"  
)  
public class CatMesCargo implements java.io.Serializable {  
  
  
 private BigDecimal mesId;  
 private BigDecimal numeroMes;  
 private Date fechaMes;  
 private Set<CatCargoFijo> catCargoFijos = new HashSet<CatCargoFijo>(0);  
  
 public CatMesCargo() {  
 }  
  
   
 public CatMesCargo(BigDecimal numeroMes, Date fechaMes) {  
 this.numeroMes = numeroMes;  
 this.fechaMes = fechaMes;  
 }  
 public CatMesCargo(BigDecimal numeroMes, Date fechaMes, Set<CatCargoFijo> catCargoFijos) {  
 this.numeroMes = numeroMes;  
 this.fechaMes = fechaMes;  
 this.catCargoFijos = catCargoFijos;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="MES\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getMesId() {  
 return this.mesId;  
 }  
   
 public void setMesId(BigDecimal mesId) {  
 this.mesId = mesId;  
 }  
  
   
 @Column(name="NUMERO\_MES", nullable=false, precision=22, scale=0)  
 public BigDecimal getNumeroMes() {  
 return this.numeroMes;  
 }  
   
 public void setNumeroMes(BigDecimal numeroMes) {  
 this.numeroMes = numeroMes;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="FECHA\_MES", nullable=false, length=7)  
 public Date getFechaMes() {  
 return this.fechaMes;  
 }  
   
 public void setFechaMes(Date fechaMes) {  
 this.fechaMes = fechaMes;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catMesCargo")  
 public Set<CatCargoFijo> getCatCargoFijos() {  
 return this.catCargoFijos;  
 }  
   
 public void setCatCargoFijos(Set<CatCargoFijo> catCargoFijos) {  
 this.catCargoFijos = catCargoFijos;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatMesDia.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.Date;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* CatMesDia generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_MES\_DIA"  
)  
public class CatMesDia implements java.io.Serializable {  
  
  
 private BigDecimal mesDiaId;  
 private CatAnioMes catAnioMes;  
 private BigDecimal dia;  
 private String descripcion;  
 private String diaSemana;  
 private BigDecimal totalHoras;  
 private BigDecimal habil;  
 private Date fechaCalendario;  
 private Set<CatMesDiaFirma> catMesDiaFirmas = new HashSet<CatMesDiaFirma>(0);  
  
 public CatMesDia() {  
 }  
  
   
 public CatMesDia(BigDecimal dia, String diaSemana, BigDecimal totalHoras, BigDecimal habil) {  
 this.dia = dia;  
 this.diaSemana = diaSemana;  
 this.totalHoras = totalHoras;  
 this.habil = habil;  
 }  
 public CatMesDia(CatAnioMes catAnioMes, BigDecimal dia, String descripcion, String diaSemana, BigDecimal totalHoras, BigDecimal habil, Date fechaCalendario, Set<CatMesDiaFirma> catMesDiaFirmas) {  
 this.catAnioMes = catAnioMes;  
 this.dia = dia;  
 this.descripcion = descripcion;  
 this.diaSemana = diaSemana;  
 this.totalHoras = totalHoras;  
 this.habil = habil;  
 this.fechaCalendario = fechaCalendario;  
 this.catMesDiaFirmas = catMesDiaFirmas;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="MES\_DIA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getMesDiaId() {  
 return this.mesDiaId;  
 }  
   
 public void setMesDiaId(BigDecimal mesDiaId) {  
 this.mesDiaId = mesDiaId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="ANIO\_MES\_ID")  
 public CatAnioMes getCatAnioMes() {  
 return this.catAnioMes;  
 }  
   
 public void setCatAnioMes(CatAnioMes catAnioMes) {  
 this.catAnioMes = catAnioMes;  
 }  
  
   
 @Column(name="DIA", nullable=false, precision=22, scale=0)  
 public BigDecimal getDia() {  
 return this.dia;  
 }  
   
 public void setDia(BigDecimal dia) {  
 this.dia = dia;  
 }  
  
   
 @Column(name="DESCRIPCION", length=1000)  
 public String getDescripcion() {  
 return this.descripcion;  
 }  
   
 public void setDescripcion(String descripcion) {  
 this.descripcion = descripcion;  
 }  
  
   
 @Column(name="DIA\_SEMANA", nullable=false, length=25)  
 public String getDiaSemana() {  
 return this.diaSemana;  
 }  
   
 public void setDiaSemana(String diaSemana) {  
 this.diaSemana = diaSemana;  
 }  
  
   
 @Column(name="TOTAL\_HORAS", nullable=false, precision=22, scale=0)  
 public BigDecimal getTotalHoras() {  
 return this.totalHoras;  
 }  
   
 public void setTotalHoras(BigDecimal totalHoras) {  
 this.totalHoras = totalHoras;  
 }  
  
   
 @Column(name="HABIL", nullable=false, precision=22, scale=0)  
 public BigDecimal getHabil() {  
 return this.habil;  
 }  
   
 public void setHabil(BigDecimal habil) {  
 this.habil = habil;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="FECHA\_CALENDARIO", length=7)  
 public Date getFechaCalendario() {  
 return this.fechaCalendario;  
 }  
   
 public void setFechaCalendario(Date fechaCalendario) {  
 this.fechaCalendario = fechaCalendario;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catMesDia")  
 public Set<CatMesDiaFirma> getCatMesDiaFirmas() {  
 return this.catMesDiaFirmas;  
 }  
   
 public void setCatMesDiaFirmas(Set<CatMesDiaFirma> catMesDiaFirmas) {  
 this.catMesDiaFirmas = catMesDiaFirmas;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatMesDiaFirma.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import java.util.Date;  
import java.util.HashSet;  
import java.util.Set;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatMesDiaFirma generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_MES\_DIA\_FIRMA"  
)  
public class CatMesDiaFirma implements java.io.Serializable {  
 private BigDecimal mesDiaFirmaId;  
 private CatMesDia catMesDia;  
 private CatFirmaCincominutal catFirmaCincominutal;  
 private Date fechaFirma;  
 private Set<TxCincominutal> txCincominutals = new HashSet<TxCincominutal>(0);  
  
 public CatMesDiaFirma() {  
 }  
  
 public CatMesDiaFirma(CatMesDia catMesDia, CatFirmaCincominutal catFirmaCincominutal, Date fechaFirma, Set<TxCincominutal> txCincominutals) {  
 this.catMesDia = catMesDia;  
 this.catFirmaCincominutal = catFirmaCincominutal;  
 this.fechaFirma = fechaFirma;  
 this.txCincominutals = txCincominutals;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="MES\_DIA\_FIRMA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getMesDiaFirmaId() {  
 return this.mesDiaFirmaId;  
 }  
   
 public void setMesDiaFirmaId(BigDecimal mesDiaFirmaId) {  
 this.mesDiaFirmaId = mesDiaFirmaId;  
 }  
  
 @ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="MES\_DIA\_ID")  
 public CatMesDia getCatMesDia() {  
 return this.catMesDia;  
 }  
   
 public void setCatMesDia(CatMesDia catMesDia) {  
 this.catMesDia = catMesDia;  
 }  
  
 @ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="FIRMA\_ID")  
 public CatFirmaCincominutal getCatFirmaCincominutal() {  
 return this.catFirmaCincominutal;  
 }  
   
 public void setCatFirmaCincominutal(CatFirmaCincominutal catFirmaCincominutal) {  
 this.catFirmaCincominutal = catFirmaCincominutal;  
 }  
  
   
 @Column(name="FECHA\_FIRMA")  
 public Date getFechaFirma() {  
 return this.fechaFirma;  
 }  
   
 public void setFechaFirma(Date fechaFirma) {  
 this.fechaFirma = fechaFirma;  
 }  
  
 @OneToMany(fetch=FetchType.LAZY, mappedBy="catMesDiaFirma")  
 public Set<TxCincominutal> getTxCincominutals() {  
 return this.txCincominutals;  
 }  
   
 public void setTxCincominutals(Set<TxCincominutal> txCincominutals) {  
 this.txCincominutals = txCincominutals;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatMetodoPago.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatMetodoPago generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_METODO\_PAGO"  
)  
public class CatMetodoPago implements java.io.Serializable {  
  
  
 private BigDecimal metodoPagoId;  
 private String nombre;  
 private Set<TxDocumentos> txDocumentoses = new HashSet<TxDocumentos>(0);  
  
 public CatMetodoPago() {  
 }  
  
 public CatMetodoPago(String nombre, Set<TxDocumentos> txDocumentoses) {  
 this.nombre = nombre;  
 this.txDocumentoses = txDocumentoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="METODO\_PAGO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getMetodoPagoId() {  
 return this.metodoPagoId;  
 }  
   
 public void setMetodoPagoId(BigDecimal metodoPagoId) {  
 this.metodoPagoId = metodoPagoId;  
 }  
  
   
 @Column(name="NOMBRE", length=200)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catMetodoPago")  
 public Set<TxDocumentos> getTxDocumentoses() {  
 return this.txDocumentoses;  
 }  
   
 public void setTxDocumentoses(Set<TxDocumentos> txDocumentoses) {  
 this.txDocumentoses = txDocumentoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatMonedas.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.Date;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* CatMonedas generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_MONEDAS"  
)  
public class CatMonedas implements java.io.Serializable {  
  
  
 private BigDecimal monedaId;  
 private String nombre;  
 private BigDecimal tipoCambio;  
 private Date fecha;  
 private Boolean activo;  
 private String clave;  
 private Boolean principal;  
 private Set<CatClienteMonedas> catClienteMonedases = new HashSet<CatClienteMonedas>(0);  
 private Set<TxDocumentos> txDocumentoses = new HashSet<TxDocumentos>(0);  
  
 public CatMonedas() {  
 }  
  
 public CatMonedas(String nombre, BigDecimal tipoCambio, Date fecha, Boolean activo, String clave, Boolean principal, Set<CatClienteMonedas> catClienteMonedases, Set<TxDocumentos> txDocumentoses) {  
 this.nombre = nombre;  
 this.tipoCambio = tipoCambio;  
 this.fecha = fecha;  
 this.activo = activo;  
 this.clave = clave;  
 this.principal = principal;  
 this.catClienteMonedases = catClienteMonedases;  
 this.txDocumentoses = txDocumentoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="MONEDA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getMonedaId() {  
 return this.monedaId;  
 }  
   
 public void setMonedaId(BigDecimal monedaId) {  
 this.monedaId = monedaId;  
 }  
  
   
 @Column(name="NOMBRE")  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="TIPO\_CAMBIO", precision=22, scale=0)  
 public BigDecimal getTipoCambio() {  
 return this.tipoCambio;  
 }  
   
 public void setTipoCambio(BigDecimal tipoCambio) {  
 this.tipoCambio = tipoCambio;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="FECHA", length=7)  
 public Date getFecha() {  
 return this.fecha;  
 }  
   
 public void setFecha(Date fecha) {  
 this.fecha = fecha;  
 }  
  
   
 @Column(name="ACTIVO", precision=1, scale=0)  
 public Boolean getActivo() {  
 return this.activo;  
 }  
   
 public void setActivo(Boolean activo) {  
 this.activo = activo;  
 }  
  
   
 @Column(name="CLAVE", length=3)  
 public String getClave() {  
 return this.clave;  
 }  
   
 public void setClave(String clave) {  
 this.clave = clave;  
 }  
  
   
 @Column(name="PRINCIPAL", precision=1, scale=0)  
 public Boolean getPrincipal() {  
 return this.principal;  
 }  
   
 public void setPrincipal(Boolean principal) {  
 this.principal = principal;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catMonedas")  
 public Set<CatClienteMonedas> getCatClienteMonedases() {  
 return this.catClienteMonedases;  
 }  
   
 public void setCatClienteMonedases(Set<CatClienteMonedas> catClienteMonedases) {  
 this.catClienteMonedases = catClienteMonedases;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catMonedas")  
 public Set<TxDocumentos> getTxDocumentoses() {  
 return this.txDocumentoses;  
 }  
   
 public void setTxDocumentoses(Set<TxDocumentos> txDocumentoses) {  
 this.txDocumentoses = txDocumentoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatOpcionPlanta.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
  
import java.math.BigDecimal;  
  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.Table;  
  
import javax.validation.constraints.NotNull;  
  
import javax.xml.bind.annotation.XmlRootElement;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_OPCION\_PLANTA")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatOpcionPlanta.findAll", query = "SELECT c FROM CatOpcionPlanta c")  
 , @NamedQuery(name = "CatOpcionPlanta.findByOpcionPlantaId", query = "SELECT c FROM CatOpcionPlanta c WHERE c.opcionPlantaId = :opcionPlantaId")})  
public class CatOpcionPlanta implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "OPCION\_PLANTA\_ID")  
 private BigDecimal opcionPlantaId;  
 @JoinColumn(name = "DETALLE\_OPCIONES\_ID", referencedColumnName = "DETALLE\_OPCIONES\_ID")  
 @ManyToOne  
 private CatDetalleOpciones detalleOpcionesId;  
 @JoinColumn(name = "PLANTA\_ID", referencedColumnName = "PLANTA\_ID")  
 @ManyToOne  
 private CatPlanta plantaId;  
  
 public CatOpcionPlanta() {  
 }  
  
 public CatOpcionPlanta(BigDecimal opcionPlantaId) {  
 this.opcionPlantaId = opcionPlantaId;  
 }  
  
 public BigDecimal getOpcionPlantaId() {  
 return opcionPlantaId;  
 }  
  
 public void setOpcionPlantaId(BigDecimal opcionPlantaId) {  
 this.opcionPlantaId = opcionPlantaId;  
 }  
  
 public CatDetalleOpciones getDetalleOpcionesId() {  
 return detalleOpcionesId;  
 }  
  
 public void setDetalleOpcionesId(CatDetalleOpciones detalleOpcionesId) {  
 this.detalleOpcionesId = detalleOpcionesId;  
 }  
  
 public CatPlanta getPlantaId() {  
 return plantaId;  
 }  
  
 public void setPlantaId(CatPlanta plantaId) {  
 this.plantaId = plantaId;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (opcionPlantaId != null ? opcionPlantaId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatOpcionPlanta)) {  
 return false;  
 }  
 CatOpcionPlanta other = (CatOpcionPlanta) object;  
 if ((this.opcionPlantaId == null && other.opcionPlantaId != null) || (this.opcionPlantaId != null && !this.opcionPlantaId.equals(other.opcionPlantaId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatOpcionPlanta[ opcionPlantaId=" + opcionPlantaId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatOpcionTag.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.Table;  
import javax.validation.constraints.NotNull;  
import javax.xml.bind.annotation.XmlRootElement;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_OPCION\_TAG")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatOpcionTag.findAll", query = "SELECT c FROM CatOpcionTag c")  
 , @NamedQuery(name = "CatOpcionTag.findByOpcionTagId", query = "SELECT c FROM CatOpcionTag c WHERE c.opcionTagId = :opcionTagId")})  
public class CatOpcionTag implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "OPCION\_TAG\_ID")  
 private BigDecimal opcionTagId;  
 @JoinColumn(name = "DETALLE\_OPCIONES\_ID", referencedColumnName = "DETALLE\_OPCIONES\_ID")  
 @ManyToOne  
 private CatDetalleOpciones detalleOpcionesId;  
 @JoinColumn(name = "TAG\_ID", referencedColumnName = "TAG\_ID")  
 @ManyToOne  
 private CatTags tagId;  
  
 public CatOpcionTag() {  
 }  
  
 public CatOpcionTag(BigDecimal opcionTagId) {  
 this.opcionTagId = opcionTagId;  
 }  
  
 public BigDecimal getOpcionTagId() {  
 return opcionTagId;  
 }  
  
 public void setOpcionTagId(BigDecimal opcionTagId) {  
 this.opcionTagId = opcionTagId;  
 }  
  
 public CatDetalleOpciones getDetalleOpcionesId() {  
 return detalleOpcionesId;  
 }  
  
 public void setDetalleOpcionesId(CatDetalleOpciones detalleOpcionesId) {  
 this.detalleOpcionesId = detalleOpcionesId;  
 }  
  
 public CatTags getTagId() {  
 return tagId;  
 }  
  
 public void setTagId(CatTags tagId) {  
 this.tagId = tagId;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (opcionTagId != null ? opcionTagId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatOpcionTag)) {  
 return false;  
 }  
 CatOpcionTag other = (CatOpcionTag) object;  
 if ((this.opcionTagId == null && other.opcionTagId != null) || (this.opcionTagId != null && !this.opcionTagId.equals(other.opcionTagId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatOpcionTag[ opcionTagId=" + opcionTagId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatOpcionTraduccion.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.Table;  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
import javax.xml.bind.annotation.XmlRootElement;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_OPCION\_TRADUCCION")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatOpcionTraduccion.findAll", query = "SELECT c FROM CatOpcionTraduccion c")  
 , @NamedQuery(name = "CatOpcionTraduccion.findByOpcionTraducionId", query = "SELECT c FROM CatOpcionTraduccion c WHERE c.opcionTraducionId = :opcionTraducionId")  
 , @NamedQuery(name = "CatOpcionTraduccion.findByTraduccion", query = "SELECT c FROM CatOpcionTraduccion c WHERE c.traduccion = :traduccion")})  
public class CatOpcionTraduccion implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "OPCION\_TRADUCION\_ID")  
 private BigDecimal opcionTraducionId;  
 @Size(max = 1040)  
 @Column(name = "TRADUCCION")  
 private String traduccion;  
 @JoinColumn(name = "DETALLE\_OPCIONES\_ID", referencedColumnName = "DETALLE\_OPCIONES\_ID")  
 @ManyToOne  
 private CatDetalleOpciones detalleOpcionesId;  
 @JoinColumn(name = "LENGUAJE\_ID", referencedColumnName = "LENGUAJE\_ID")  
 @ManyToOne  
 private CatLenguajes lenguajeId;  
  
 public CatOpcionTraduccion() {  
 }  
  
 public CatOpcionTraduccion(BigDecimal opcionTraducionId) {  
 this.opcionTraducionId = opcionTraducionId;  
 }  
  
 public BigDecimal getOpcionTraducionId() {  
 return opcionTraducionId;  
 }  
  
 public void setOpcionTraducionId(BigDecimal opcionTraducionId) {  
 this.opcionTraducionId = opcionTraducionId;  
 }  
  
 public String getTraduccion() {  
 return traduccion;  
 }  
  
 public void setTraduccion(String traduccion) {  
 this.traduccion = traduccion;  
 }  
  
 public CatDetalleOpciones getDetalleOpcionesId() {  
 return detalleOpcionesId;  
 }  
  
 public void setDetalleOpcionesId(CatDetalleOpciones detalleOpcionesId) {  
 this.detalleOpcionesId = detalleOpcionesId;  
 }  
  
 public CatLenguajes getLenguajeId() {  
 return lenguajeId;  
 }  
  
 public void setLenguajeId(CatLenguajes lenguajeId) {  
 this.lenguajeId = lenguajeId;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (opcionTraducionId != null ? opcionTraducionId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatOpcionTraduccion)) {  
 return false;  
 }  
 CatOpcionTraduccion other = (CatOpcionTraduccion) object;  
 if ((this.opcionTraducionId == null && other.opcionTraducionId != null) || (this.opcionTraducionId != null && !this.opcionTraducionId.equals(other.opcionTraducionId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatOpcionTraduccion[ opcionTraducionId=" + opcionTraducionId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatPaises.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatPaises generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_PAISES"  
)  
public class CatPaises implements java.io.Serializable {  
  
  
 private BigDecimal paisId;  
 private String nombre;  
 private Boolean activo;  
 private String claveSat;  
 private Set<CatEstados> catEstadoses = new HashSet<CatEstados>(0);  
  
 public CatPaises() {  
 }  
  
 public CatPaises(String nombre, Boolean activo, String claveSat, Set<CatEstados> catEstadoses) {  
 this.nombre = nombre;  
 this.activo = activo;  
 this.claveSat = claveSat;  
 this.catEstadoses = catEstadoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="PAIS\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getPaisId() {  
 return this.paisId;  
 }  
   
 public void setPaisId(BigDecimal paisId) {  
 this.paisId = paisId;  
 }  
  
   
 @Column(name="NOMBRE")  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="ACTIVO", precision=1, scale=0)  
 public Boolean getActivo() {  
 return this.activo;  
 }  
   
 public void setActivo(Boolean activo) {  
 this.activo = activo;  
 }  
  
   
 @Column(name="CLAVE\_SAT", length=10)  
 public String getClaveSat() {  
 return this.claveSat;  
 }  
   
 public void setClaveSat(String claveSat) {  
 this.claveSat = claveSat;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catPaises")  
 public Set<CatEstados> getCatEstadoses() {  
 return this.catEstadoses;  
 }  
   
 public void setCatEstadoses(Set<CatEstados> catEstadoses) {  
 this.catEstadoses = catEstadoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatPermiso.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatPermiso generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_PERMISO"  
)  
public class CatPermiso implements java.io.Serializable {  
  
  
 private BigDecimal permisoId;  
 private String permiso;  
 private Set<TxRolPermiso> txRolPermisos = new HashSet<TxRolPermiso>(0);  
  
 public CatPermiso() {  
 }  
  
   
 public CatPermiso(String permiso) {  
 this.permiso = permiso;  
 }  
 public CatPermiso(String permiso, Set<TxRolPermiso> txRolPermisos) {  
 this.permiso = permiso;  
 this.txRolPermisos = txRolPermisos;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="PERMISO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getPermisoId() {  
 return this.permisoId;  
 }  
   
 public void setPermisoId(BigDecimal permisoId) {  
 this.permisoId = permisoId;  
 }  
  
   
 @Column(name="PERMISO", nullable=false, length=100)  
 public String getPermiso() {  
 return this.permiso;  
 }  
   
 public void setPermiso(String permiso) {  
 this.permiso = permiso;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catPermiso")  
 public Set<TxRolPermiso> getTxRolPermisos() {  
 return this.txRolPermisos;  
 }  
   
 public void setTxRolPermisos(Set<TxRolPermiso> txRolPermisos) {  
 this.txRolPermisos = txRolPermisos;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatPlanta.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import java.util.HashSet;  
import java.util.Set;  
  
import javax.persistence.CascadeType;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatPlanta generated by hbm2java  
 \*/  
@Entity  
@Table(name = "CAT\_PLANTA")  
public class CatPlanta implements java.io.Serializable {  
  
  
 private BigDecimal plantaId;  
 private String nombre;  
 private String siglas;  
 private Set<CatLimitesDominio> catLimitesDominios = new HashSet<CatLimitesDominio>(0);  
 private Set<CatAnioMes> catAnioMeses = new HashSet<CatAnioMes>(0);  
 private Set<CatCargos> catCargoses = new HashSet<CatCargos>(0);  
 private Set<CatCurvasCorreccion> catCurvasCorreccions = new HashSet<CatCurvasCorreccion>(0);  
 private Set<CatVariable> catVariables = new HashSet<CatVariable>(0);  
 private Set<CatRangoVariable> catRangoVariables = new HashSet<CatRangoVariable>(0);  
 private Set<TxCincominutal> txCincominutals = new HashSet<TxCincominutal>(0);  
 private Set<TxUsuarioPlanta> txUsuarioPlantas = new HashSet<TxUsuarioPlanta>(0);  
 private Set<CatPlantaTipoFacturas> catPlantaTipoFacturases = new HashSet<CatPlantaTipoFacturas>(0);  
  
 private Set<TxBitacora> txBitacoraSet;  
 private Set<CatVariablePerfil> catVariablePerfils = new HashSet<CatVariablePerfil>(0);  
 private Set<CatCompaniaPlantas> catCompaniaPlantases = new HashSet<CatCompaniaPlantas>(0);  
 private Set<CatMedidores> catMedidoreses = new HashSet<CatMedidores>(0);  
 private Set<CatCondiciones> catCondicioneses = new HashSet<CatCondiciones>(0);  
  
 public CatPlanta() {  
 }  
  
  
 public CatPlanta(String nombre) {  
 this.nombre = nombre;  
 }  
  
 public CatPlanta(String nombre, String siglas, Set<CatLimitesDominio> catLimitesDominios,  
 Set<CatAnioMes> catAnioMeses, Set<CatCargos> catCargoses,  
 Set<CatCurvasCorreccion> catCurvasCorreccions, Set<CatVariable> catVariables,  
 Set<CatRangoVariable> catRangoVariables, Set<TxCincominutal> txCincominutals,  
 Set<TxUsuarioPlanta> txUsuarioPlantas, Set<CatPlantaTipoFacturas> catPlantaTipoFacturases,  
 Set<TxBitacora> txBitacoras, Set<CatVariablePerfil> catVariablePerfils,  
 Set<CatCompaniaPlantas> catCompaniaPlantases, Set<CatMedidores> catMedidoreses,  
 Set<CatCondiciones> catCondicioneses) {  
 this.nombre = nombre;  
 this.siglas = siglas;  
 this.catLimitesDominios = catLimitesDominios;  
 this.catAnioMeses = catAnioMeses;  
 this.catCargoses = catCargoses;  
 this.catCurvasCorreccions = catCurvasCorreccions;  
 this.catVariables = catVariables;  
 this.catRangoVariables = catRangoVariables;  
 this.txCincominutals = txCincominutals;  
 this.txUsuarioPlantas = txUsuarioPlantas;  
 this.catPlantaTipoFacturases = catPlantaTipoFacturases;  
 this.txBitacoraSet = txBitacoras;  
 this.catVariablePerfils = catVariablePerfils;  
 this.catCompaniaPlantases = catCompaniaPlantases;  
 this.catMedidoreses = catMedidoreses;  
 this.catCondicioneses = catCondicioneses;  
 }  
  
 @Id  
 @GeneratedValue(strategy = IDENTITY)  
  
  
 @Column(name = "PLANTA\_ID", unique = true, nullable = false, precision = 22, scale = 0)  
 public BigDecimal getPlantaId() {  
 return this.plantaId;  
 }  
  
 public void setPlantaId(BigDecimal plantaId) {  
 this.plantaId = plantaId;  
 }  
  
  
 @Column(name = "NOMBRE", nullable = false, length = 100)  
 public String getNombre() {  
 return this.nombre;  
 }  
  
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
  
 @Column(name = "SIGLAS", length = 50)  
 public String getSiglas() {  
 return this.siglas;  
 }  
  
 public void setSiglas(String siglas) {  
 this.siglas = siglas;  
 }  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "catPlanta")  
 public Set<CatLimitesDominio> getCatLimitesDominios() {  
 return this.catLimitesDominios;  
 }  
  
 public void setCatLimitesDominios(Set<CatLimitesDominio> catLimitesDominios) {  
 this.catLimitesDominios = catLimitesDominios;  
 }  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "catPlanta")  
 public Set<CatAnioMes> getCatAnioMeses() {  
 return this.catAnioMeses;  
 }  
  
 public void setCatAnioMeses(Set<CatAnioMes> catAnioMeses) {  
 this.catAnioMeses = catAnioMeses;  
 }  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "catPlanta")  
 public Set<CatCargos> getCatCargoses() {  
 return this.catCargoses;  
 }  
  
 public void setCatCargoses(Set<CatCargos> catCargoses) {  
 this.catCargoses = catCargoses;  
 }  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "catPlanta")  
 public Set<CatCurvasCorreccion> getCatCurvasCorreccions() {  
 return this.catCurvasCorreccions;  
 }  
  
 public void setCatCurvasCorreccions(Set<CatCurvasCorreccion> catCurvasCorreccions) {  
 this.catCurvasCorreccions = catCurvasCorreccions;  
 }  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "catPlanta")  
 public Set<CatVariable> getCatVariables() {  
 return this.catVariables;  
 }  
  
 public void setCatVariables(Set<CatVariable> catVariables) {  
 this.catVariables = catVariables;  
 }  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "catPlanta")  
 public Set<CatRangoVariable> getCatRangoVariables() {  
 return this.catRangoVariables;  
 }  
  
 public void setCatRangoVariables(Set<CatRangoVariable> catRangoVariables) {  
 this.catRangoVariables = catRangoVariables;  
 }  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "catPlanta")  
 public Set<TxCincominutal> getTxCincominutals() {  
 return this.txCincominutals;  
 }  
  
 public void setTxCincominutals(Set<TxCincominutal> txCincominutals) {  
 this.txCincominutals = txCincominutals;  
 }  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "catPlanta")  
 public Set<TxUsuarioPlanta> getTxUsuarioPlantas() {  
 return this.txUsuarioPlantas;  
 }  
  
 public void setTxUsuarioPlantas(Set<TxUsuarioPlanta> txUsuarioPlantas) {  
 this.txUsuarioPlantas = txUsuarioPlantas;  
 }  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "catPlanta")  
 public Set<CatPlantaTipoFacturas> getCatPlantaTipoFacturases() {  
 return this.catPlantaTipoFacturases;  
 }  
  
 public void setCatPlantaTipoFacturases(Set<CatPlantaTipoFacturas> catPlantaTipoFacturases) {  
 this.catPlantaTipoFacturases = catPlantaTipoFacturases;  
 }  
  
 @OneToMany(cascade = CascadeType.ALL, mappedBy = "plantaId")  
 public Set<TxBitacora> getTxBitacoraSet() {  
 return txBitacoraSet;  
 }  
  
 public void setTxBitacoraSet(Set<TxBitacora> txBitacoraSet) {  
 this.txBitacoraSet = txBitacoraSet;  
 }  
  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "catPlanta")  
 public Set<CatVariablePerfil> getCatVariablePerfils() {  
 return this.catVariablePerfils;  
 }  
  
 public void setCatVariablePerfils(Set<CatVariablePerfil> catVariablePerfils) {  
 this.catVariablePerfils = catVariablePerfils;  
 }  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "catPlanta")  
 public Set<CatCompaniaPlantas> getCatCompaniaPlantases() {  
 return this.catCompaniaPlantases;  
 }  
  
 public void setCatCompaniaPlantases(Set<CatCompaniaPlantas> catCompaniaPlantases) {  
 this.catCompaniaPlantases = catCompaniaPlantases;  
 }  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "catPlanta")  
 public Set<CatMedidores> getCatMedidoreses() {  
 return this.catMedidoreses;  
 }  
  
 public void setCatMedidoreses(Set<CatMedidores> catMedidoreses) {  
 this.catMedidoreses = catMedidoreses;  
 }  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "catPlanta")  
 public Set<CatCondiciones> getCatCondicioneses() {  
 return this.catCondicioneses;  
 }  
  
 public void setCatCondicioneses(Set<CatCondiciones> catCondicioneses) {  
 this.catCondicioneses = catCondicioneses;  
 }  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatPlantaTipoFacturas.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatPlantaTipoFacturas generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_PLANTA\_TIPO\_FACTURAS"  
)  
public class CatPlantaTipoFacturas implements java.io.Serializable {  
  
  
 private BigDecimal plantaTipoFacturaId;  
 private CatPlanta catPlanta;  
 private CatTiposFacturas catTiposFacturas;  
  
 public CatPlantaTipoFacturas() {  
 }  
  
 public CatPlantaTipoFacturas(CatPlanta catPlanta, CatTiposFacturas catTiposFacturas) {  
 this.catPlanta = catPlanta;  
 this.catTiposFacturas = catTiposFacturas;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="PLANTA\_TIPO\_FACTURA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getPlantaTipoFacturaId() {  
 return this.plantaTipoFacturaId;  
 }  
   
 public void setPlantaTipoFacturaId(BigDecimal plantaTipoFacturaId) {  
 this.plantaTipoFacturaId = plantaTipoFacturaId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="PLANTA\_ID")  
 public CatPlanta getCatPlanta() {  
 return this.catPlanta;  
 }  
   
 public void setCatPlanta(CatPlanta catPlanta) {  
 this.catPlanta = catPlanta;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="TIPO\_FACTURA\_ID")  
 public CatTiposFacturas getCatTiposFacturas() {  
 return this.catTiposFacturas;  
 }  
   
 public void setCatTiposFacturas(CatTiposFacturas catTiposFacturas) {  
 this.catTiposFacturas = catTiposFacturas;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatPropCincominutal.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatPropCincominutal generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_PROP\_CINCOMINUTAL"  
)  
public class CatPropCincominutal implements java.io.Serializable {  
  
  
 private BigDecimal propiedadId;  
 private String propiedad;  
 private Set<TxProduccion> txProduccions = new HashSet<TxProduccion>(0);  
  
 public CatPropCincominutal() {  
 }  
  
   
 public CatPropCincominutal(String propiedad) {  
 this.propiedad = propiedad;  
 }  
 public CatPropCincominutal(String propiedad, Set<TxProduccion> txProduccions) {  
 this.propiedad = propiedad;  
 this.txProduccions = txProduccions;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="PROPIEDAD\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getPropiedadId() {  
 return this.propiedadId;  
 }  
   
 public void setPropiedadId(BigDecimal propiedadId) {  
 this.propiedadId = propiedadId;  
 }  
  
   
 @Column(name="PROPIEDAD", nullable=false, length=150)  
 public String getPropiedad() {  
 return this.propiedad;  
 }  
   
 public void setPropiedad(String propiedad) {  
 this.propiedad = propiedad;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catPropCincominutal")  
 public Set<TxProduccion> getTxProduccions() {  
 return this.txProduccions;  
 }  
   
 public void setTxProduccions(Set<TxProduccion> txProduccions) {  
 this.txProduccions = txProduccions;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatQuery.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatQuery generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_QUERY"  
)  
public class CatQuery implements java.io.Serializable {  
  
  
 private BigDecimal queryId;  
 private String query;  
 private Set<TxTrazaQuery> txTrazaQueries = new HashSet<TxTrazaQuery>(0);  
  
 public CatQuery() {  
 }  
  
   
 public CatQuery(String query) {  
 this.query = query;  
 }  
 public CatQuery(String query, Set<TxTrazaQuery> txTrazaQueries) {  
 this.query = query;  
 this.txTrazaQueries = txTrazaQueries;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="QUERY\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getQueryId() {  
 return this.queryId;  
 }  
   
 public void setQueryId(BigDecimal queryId) {  
 this.queryId = queryId;  
 }  
  
   
 @Column(name="QUERY", nullable=false, length=500)  
 public String getQuery() {  
 return this.query;  
 }  
   
 public void setQuery(String query) {  
 this.query = query;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catQuery")  
 public Set<TxTrazaQuery> getTxTrazaQueries() {  
 return this.txTrazaQueries;  
 }  
   
 public void setTxTrazaQueries(Set<TxTrazaQuery> txTrazaQueries) {  
 this.txTrazaQueries = txTrazaQueries;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatRangoVariable.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatRangoVariable generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_RANGO\_VARIABLE"  
)  
public class CatRangoVariable implements java.io.Serializable {  
  
  
 private BigDecimal rangoId;  
 private CatPlanta catPlanta;  
 private CatVariable catVariable;  
 private BigDecimal valorMinimo;  
 private BigDecimal valorMaximo;  
 private BigDecimal diferenciaMaxima;  
 private String unificacionEscala;  
  
 public CatRangoVariable() {  
 }  
  
   
 public CatRangoVariable(CatPlanta catPlanta, CatVariable catVariable) {  
 this.catPlanta = catPlanta;  
 this.catVariable = catVariable;  
 }  
 public CatRangoVariable(CatPlanta catPlanta, CatVariable catVariable, BigDecimal valorMinimo, BigDecimal valorMaximo, BigDecimal diferenciaMaxima, String unificacionEscala) {  
 this.catPlanta = catPlanta;  
 this.catVariable = catVariable;  
 this.valorMinimo = valorMinimo;  
 this.valorMaximo = valorMaximo;  
 this.diferenciaMaxima = diferenciaMaxima;  
 this.unificacionEscala = unificacionEscala;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="RANGO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getRangoId() {  
 return this.rangoId;  
 }  
   
 public void setRangoId(BigDecimal rangoId) {  
 this.rangoId = rangoId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="PLANTA\_ID", nullable=false)  
 public CatPlanta getCatPlanta() {  
 return this.catPlanta;  
 }  
   
 public void setCatPlanta(CatPlanta catPlanta) {  
 this.catPlanta = catPlanta;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="VARIABLE\_ID", nullable=false)  
 public CatVariable getCatVariable() {  
 return this.catVariable;  
 }  
   
 public void setCatVariable(CatVariable catVariable) {  
 this.catVariable = catVariable;  
 }  
  
   
 @Column(name="VALOR\_MINIMO", precision=22, scale=0)  
 public BigDecimal getValorMinimo() {  
 return this.valorMinimo;  
 }  
   
 public void setValorMinimo(BigDecimal valorMinimo) {  
 this.valorMinimo = valorMinimo;  
 }  
  
   
 @Column(name="VALOR\_MAXIMO", precision=22, scale=0)  
 public BigDecimal getValorMaximo() {  
 return this.valorMaximo;  
 }  
   
 public void setValorMaximo(BigDecimal valorMaximo) {  
 this.valorMaximo = valorMaximo;  
 }  
  
   
 @Column(name="DIFERENCIA\_MAXIMA", precision=22, scale=0)  
 public BigDecimal getDiferenciaMaxima() {  
 return this.diferenciaMaxima;  
 }  
   
 public void setDiferenciaMaxima(BigDecimal diferenciaMaxima) {  
 this.diferenciaMaxima = diferenciaMaxima;  
 }  
  
   
 @Column(name="UNIFICACION\_ESCALA", length=100)  
 public String getUnificacionEscala() {  
 return this.unificacionEscala;  
 }  
   
 public void setUnificacionEscala(String unificacionEscala) {  
 this.unificacionEscala = unificacionEscala;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatRegimenFiscal.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import java.util.HashSet;  
import java.util.Set;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatRegimenFiscal generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_REGIMEN\_FISCAL"  
)  
public class CatRegimenFiscal implements java.io.Serializable {  
  
  
 private BigDecimal regimenFiscalId;  
 private String nombre;  
 private String claveNumero;  
 private int paraPersonasFisicas;  
 private int paraPersonasMorales;  
 private Set<CatDatosFiscales> catDatosFiscaleses = new HashSet<CatDatosFiscales>(0);  
  
 public CatRegimenFiscal() {  
 }  
  
 public CatRegimenFiscal(String nombre, String claveNumero, Set<CatDatosFiscales> catDatosFiscaleses) {  
 this.nombre = nombre;  
 this.claveNumero = claveNumero;  
 this.catDatosFiscaleses = catDatosFiscaleses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="REGIMEN\_FISCAL\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getRegimenFiscalId() {  
 return this.regimenFiscalId;  
 }  
   
 public void setRegimenFiscalId(BigDecimal regimenFiscalId) {  
 this.regimenFiscalId = regimenFiscalId;  
 }  
  
   
 @Column(name="NOMBRE")  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
  
 public void setParaPersonasFisicas(int paraPersonasFisicas) {  
 this.paraPersonasFisicas = paraPersonasFisicas;  
 }  
 @Column(name="PARA\_PERSONAS\_FISICAS")  
 public int getParaPersonasFisicas() {  
 return paraPersonasFisicas;  
 }  
  
 public void setParaPersonasMorales(int paraPersonasMorales) {  
 this.paraPersonasMorales = paraPersonasMorales;  
 }  
 @Column(name="PARA\_PERSONAS\_MORALES")  
 public int getParaPersonasMorales() {  
 return paraPersonasMorales;  
 }  
  
 @Column(name="CLAVE\_NUMERO", length=10)  
 public String getClaveNumero() {  
 return this.claveNumero;  
 }  
   
 public void setClaveNumero(String claveNumero) {  
 this.claveNumero = claveNumero;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catRegimenFiscal")  
 public Set<CatDatosFiscales> getCatDatosFiscaleses() {  
 return this.catDatosFiscaleses;  
 }  
   
 public void setCatDatosFiscaleses(Set<CatDatosFiscales> catDatosFiscaleses) {  
 this.catDatosFiscaleses = catDatosFiscaleses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatReglaNegocio.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatReglaNegocio generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_REGLA\_NEGOCIO"  
)  
public class CatReglaNegocio implements java.io.Serializable {  
  
  
 private BigDecimal reglaNegocioId;  
 private CatVariable catVariable;  
 private BigDecimal estatusVarId;  
 private String algoritmo;  
 private String condicion;  
  
 public CatReglaNegocio() {  
 }  
  
 public CatReglaNegocio(CatVariable catVariable, BigDecimal estatusVarId, String algoritmo, String condicion) {  
 this.catVariable = catVariable;  
 this.estatusVarId = estatusVarId;  
 this.algoritmo = algoritmo;  
 this.condicion = condicion;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="REGLA\_NEGOCIO\_ID", nullable=false, precision=22, scale=0)  
 public BigDecimal getReglaNegocioId() {  
 return this.reglaNegocioId;  
 }  
   
 public void setReglaNegocioId(BigDecimal reglaNegocioId) {  
 this.reglaNegocioId = reglaNegocioId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="VARIABLE\_ID")  
 public CatVariable getCatVariable() {  
 return this.catVariable;  
 }  
   
 public void setCatVariable(CatVariable catVariable) {  
 this.catVariable = catVariable;  
 }  
  
   
 @Column(name="ESTATUS\_VAR\_ID", precision=22, scale=0)  
 public BigDecimal getEstatusVarId() {  
 return this.estatusVarId;  
 }  
   
 public void setEstatusVarId(BigDecimal estatusVarId) {  
 this.estatusVarId = estatusVarId;  
 }  
  
   
 @Column(name="ALGORITMO", length=250)  
 public String getAlgoritmo() {  
 return this.algoritmo;  
 }  
   
 public void setAlgoritmo(String algoritmo) {  
 this.algoritmo = algoritmo;  
 }  
  
   
 @Column(name="CONDICION", length=500)  
 public String getCondicion() {  
 return this.condicion;  
 }  
   
 public void setCondicion(String condicion) {  
 this.condicion = condicion;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatRol.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatRol generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_ROL"  
)  
public class CatRol implements java.io.Serializable {  
  
  
 private BigDecimal rolId;  
 private String rol;  
 private Set<TxRolPermiso> txRolPermisos = new HashSet<TxRolPermiso>(0);  
 private Set<TxUsuarioRol> txUsuarioRols = new HashSet<TxUsuarioRol>(0);  
  
 public CatRol() {  
 }  
  
   
 public CatRol(String rol) {  
 this.rol = rol;  
 }  
 public CatRol(String rol, Set<TxRolPermiso> txRolPermisos, Set<TxUsuarioRol> txUsuarioRols) {  
 this.rol = rol;  
 this.txRolPermisos = txRolPermisos;  
 this.txUsuarioRols = txUsuarioRols;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="ROL\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getRolId() {  
 return this.rolId;  
 }  
   
 public void setRolId(BigDecimal rolId) {  
 this.rolId = rolId;  
 }  
  
   
 @Column(name="ROL", nullable=false, length=100)  
 public String getRol() {  
 return this.rol;  
 }  
   
 public void setRol(String rol) {  
 this.rol = rol;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catRol")  
 public Set<TxRolPermiso> getTxRolPermisos() {  
 return this.txRolPermisos;  
 }  
   
 public void setTxRolPermisos(Set<TxRolPermiso> txRolPermisos) {  
 this.txRolPermisos = txRolPermisos;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catRol")  
 public Set<TxUsuarioRol> getTxUsuarioRols() {  
 return this.txUsuarioRols;  
 }  
   
 public void setTxUsuarioRols(Set<TxUsuarioRol> txUsuarioRols) {  
 this.txUsuarioRols = txUsuarioRols;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatServicioImpuestos.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatServicioImpuestos generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_SERVICIO\_IMPUESTOS"  
)  
public class CatServicioImpuestos implements java.io.Serializable {  
  
  
 private BigDecimal servicioImpuestoId;  
 private CatImpuestos catImpuestos;  
 private CatServicios catServicios;  
  
 public CatServicioImpuestos() {  
 }  
  
 public CatServicioImpuestos(CatImpuestos catImpuestos, CatServicios catServicios) {  
 this.catImpuestos = catImpuestos;  
 this.catServicios = catServicios;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="SERVICIO\_IMPUESTO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getServicioImpuestoId() {  
 return this.servicioImpuestoId;  
 }  
   
 public void setServicioImpuestoId(BigDecimal servicioImpuestoId) {  
 this.servicioImpuestoId = servicioImpuestoId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="IMPUESTO\_ID")  
 public CatImpuestos getCatImpuestos() {  
 return this.catImpuestos;  
 }  
   
 public void setCatImpuestos(CatImpuestos catImpuestos) {  
 this.catImpuestos = catImpuestos;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="SERVICIO\_ID")  
 public CatServicios getCatServicios() {  
 return this.catServicios;  
 }  
   
 public void setCatServicios(CatServicios catServicios) {  
 this.catServicios = catServicios;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatServicios.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatServicios generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_SERVICIOS"  
)  
public class CatServicios implements java.io.Serializable {  
  
  
 private BigDecimal servicioId;  
 private SatProductosUnidades satProductosUnidades;  
 private SatProductosClaves satProductosClaves;  
 private CatTiposServicios catTiposServicios;  
 private String nombre;  
 private String clave;  
 private String observaciones;  
 private String numeroIdentificacion;  
 private String descripcion;  
 private Set<CatFacturaProductos> catFacturaProductoses = new HashSet<CatFacturaProductos>(0);  
 private Set<CatServicioImpuestos> catServicioImpuestoses = new HashSet<CatServicioImpuestos>(0);  
  
 public CatServicios() {  
 }  
  
 public CatServicios(SatProductosUnidades satProductosUnidades, SatProductosClaves satProductosClaves, CatTiposServicios catTiposServicios, String nombre, String clave, String observaciones, String numeroIdentificacion, String descripcion, Set<CatFacturaProductos> catFacturaProductoses, Set<CatServicioImpuestos> catServicioImpuestoses) {  
 this.satProductosUnidades = satProductosUnidades;  
 this.satProductosClaves = satProductosClaves;  
 this.catTiposServicios = catTiposServicios;  
 this.nombre = nombre;  
 this.clave = clave;  
 this.observaciones = observaciones;  
 this.numeroIdentificacion = numeroIdentificacion;  
 this.descripcion = descripcion;  
 this.catFacturaProductoses = catFacturaProductoses;  
 this.catServicioImpuestoses = catServicioImpuestoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="SERVICIO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getServicioId() {  
 return this.servicioId;  
 }  
   
 public void setServicioId(BigDecimal servicioId) {  
 this.servicioId = servicioId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="SAT\_PRODUCTOS\_UNIDADES\_ID")  
 public SatProductosUnidades getSatProductosUnidades() {  
 return this.satProductosUnidades;  
 }  
   
 public void setSatProductosUnidades(SatProductosUnidades satProductosUnidades) {  
 this.satProductosUnidades = satProductosUnidades;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="SAT\_PRODUCTOS\_CLAVES\_ID")  
 public SatProductosClaves getSatProductosClaves() {  
 return this.satProductosClaves;  
 }  
   
 public void setSatProductosClaves(SatProductosClaves satProductosClaves) {  
 this.satProductosClaves = satProductosClaves;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="TIPO\_SERVICIO\_ID")  
 public CatTiposServicios getCatTiposServicios() {  
 return this.catTiposServicios;  
 }  
   
 public void setCatTiposServicios(CatTiposServicios catTiposServicios) {  
 this.catTiposServicios = catTiposServicios;  
 }  
  
   
 @Column(name="NOMBRE")  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="CLAVE")  
 public String getClave() {  
 return this.clave;  
 }  
   
 public void setClave(String clave) {  
 this.clave = clave;  
 }  
  
   
 @Column(name="OBSERVACIONES")  
 public String getObservaciones() {  
 return this.observaciones;  
 }  
   
 public void setObservaciones(String observaciones) {  
 this.observaciones = observaciones;  
 }  
  
   
 @Column(name="NUMERO\_IDENTIFICACION")  
 public String getNumeroIdentificacion() {  
 return this.numeroIdentificacion;  
 }  
   
 public void setNumeroIdentificacion(String numeroIdentificacion) {  
 this.numeroIdentificacion = numeroIdentificacion;  
 }  
  
   
 @Column(name="DESCRIPCION")  
 public String getDescripcion() {  
 return this.descripcion;  
 }  
   
 public void setDescripcion(String descripcion) {  
 this.descripcion = descripcion;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catServicios")  
 public Set<CatFacturaProductos> getCatFacturaProductoses() {  
 return this.catFacturaProductoses;  
 }  
   
 public void setCatFacturaProductoses(Set<CatFacturaProductos> catFacturaProductoses) {  
 this.catFacturaProductoses = catFacturaProductoses;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catServicios")  
 public Set<CatServicioImpuestos> getCatServicioImpuestoses() {  
 return this.catServicioImpuestoses;  
 }  
   
 public void setCatServicioImpuestoses(Set<CatServicioImpuestos> catServicioImpuestoses) {  
 this.catServicioImpuestoses = catServicioImpuestoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatSistemas.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import java.math.BigInteger;  
import java.util.Date;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
import javax.xml.bind.annotation.XmlRootElement;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_SISTEMAS")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatSistemas.findAll", query = "SELECT c FROM CatSistemas c")  
 , @NamedQuery(name = "CatSistemas.findBySistemaId", query = "SELECT c FROM CatSistemas c WHERE c.sistemaId = :sistemaId")  
 , @NamedQuery(name = "CatSistemas.findByNombre", query = "SELECT c FROM CatSistemas c WHERE c.nombre = :nombre")  
 , @NamedQuery(name = "CatSistemas.findByActivo", query = "SELECT c FROM CatSistemas c WHERE c.activo = :activo")  
 , @NamedQuery(name = "CatSistemas.findByFechaRegistro", query = "SELECT c FROM CatSistemas c WHERE c.fechaRegistro = :fechaRegistro")  
 , @NamedQuery(name = "CatSistemas.findByUsuarioRegistro", query = "SELECT c FROM CatSistemas c WHERE c.usuarioRegistro = :usuarioRegistro")  
 , @NamedQuery(name = "CatSistemas.findByFechaModificacion", query = "SELECT c FROM CatSistemas c WHERE c.fechaModificacion = :fechaModificacion")  
 , @NamedQuery(name = "CatSistemas.findByUsuarioModificoId", query = "SELECT c FROM CatSistemas c WHERE c.usuarioModificoId = :usuarioModificoId")})  
public class CatSistemas implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "SISTEMA\_ID")  
 private BigDecimal sistemaId;  
 @Size(max = 1020)  
 @Column(name = "NOMBRE")  
 private String nombre;  
 @Column(name = "ACTIVO")  
 private BigInteger activo;  
 @Column(name = "FECHA\_REGISTRO")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaRegistro;  
 @Column(name = "USUARIO\_REGISTRO")  
 private BigInteger usuarioRegistro;  
 @Column(name = "FECHA\_MODIFICACION")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaModificacion;  
 @Column(name = "USUARIO\_MODIFICO\_ID")  
 private BigInteger usuarioModificoId;  
  
 public CatSistemas() {  
 }  
  
 public CatSistemas(BigDecimal sistemaId) {  
 this.sistemaId = sistemaId;  
 }  
  
 public BigDecimal getSistemaId() {  
 return sistemaId;  
 }  
  
 public void setSistemaId(BigDecimal sistemaId) {  
 this.sistemaId = sistemaId;  
 }  
  
 public String getNombre() {  
 return nombre;  
 }  
  
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
 public BigInteger getActivo() {  
 return activo;  
 }  
  
 public void setActivo(BigInteger activo) {  
 this.activo = activo;  
 }  
  
 public Date getFechaRegistro() {  
 return fechaRegistro;  
 }  
  
 public void setFechaRegistro(Date fechaRegistro) {  
 this.fechaRegistro = fechaRegistro;  
 }  
  
 public BigInteger getUsuarioRegistro() {  
 return usuarioRegistro;  
 }  
  
 public void setUsuarioRegistro(BigInteger usuarioRegistro) {  
 this.usuarioRegistro = usuarioRegistro;  
 }  
  
 public Date getFechaModificacion() {  
 return fechaModificacion;  
 }  
  
 public void setFechaModificacion(Date fechaModificacion) {  
 this.fechaModificacion = fechaModificacion;  
 }  
  
 public BigInteger getUsuarioModificoId() {  
 return usuarioModificoId;  
 }  
  
 public void setUsuarioModificoId(BigInteger usuarioModificoId) {  
 this.usuarioModificoId = usuarioModificoId;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (sistemaId != null ? sistemaId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatSistemas)) {  
 return false;  
 }  
 CatSistemas other = (CatSistemas) object;  
 if ((this.sistemaId == null && other.sistemaId != null) || (this.sistemaId != null && !this.sistemaId.equals(other.sistemaId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatSistemas[ sistemaId=" + sistemaId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatSubCargos.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatSubCargos generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_SUB\_CARGOS"  
)  
public class CatSubCargos implements java.io.Serializable {  
  
  
 private BigDecimal subCargoId;  
 private CatCargos catCargos;  
 private String subCargo;  
 private Set<CatCargoFijo> catCargoFijos = new HashSet<CatCargoFijo>(0);  
  
 public CatSubCargos() {  
 }  
  
   
 public CatSubCargos(CatCargos catCargos, String subCargo) {  
 this.catCargos = catCargos;  
 this.subCargo = subCargo;  
 }  
 public CatSubCargos(CatCargos catCargos, String subCargo, Set<CatCargoFijo> catCargoFijos) {  
 this.catCargos = catCargos;  
 this.subCargo = subCargo;  
 this.catCargoFijos = catCargoFijos;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="SUB\_CARGO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getSubCargoId() {  
 return this.subCargoId;  
 }  
   
 public void setSubCargoId(BigDecimal subCargoId) {  
 this.subCargoId = subCargoId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="CARGO\_ID", nullable=false)  
 public CatCargos getCatCargos() {  
 return this.catCargos;  
 }  
   
 public void setCatCargos(CatCargos catCargos) {  
 this.catCargos = catCargos;  
 }  
  
   
 @Column(name="SUB\_CARGO", nullable=false, length=100)  
 public String getSubCargo() {  
 return this.subCargo;  
 }  
   
 public void setSubCargo(String subCargo) {  
 this.subCargo = subCargo;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catSubCargos")  
 public Set<CatCargoFijo> getCatCargoFijos() {  
 return this.catCargoFijos;  
 }  
   
 public void setCatCargoFijos(Set<CatCargoFijo> catCargoFijos) {  
 this.catCargoFijos = catCargoFijos;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatTags.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
  
import java.math.BigDecimal;  
import java.math.BigInteger;  
  
import java.util.Date;  
import java.util.Set;  
  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlTransient;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_TAGS")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatTags.findAll", query = "SELECT c FROM CatTags c")  
 , @NamedQuery(name = "CatTags.findByTagId", query = "SELECT c FROM CatTags c WHERE c.tagId = :tagId")  
 , @NamedQuery(name = "CatTags.findByPlantaId", query = "SELECT c FROM CatTags c WHERE c.plantaId = :plantaId")  
 , @NamedQuery(name = "CatTags.findByTag", query = "SELECT c FROM CatTags c WHERE c.tag = :tag")  
 , @NamedQuery(name = "CatTags.findByTagDescripcion", query = "SELECT c FROM CatTags c WHERE c.tagDescripcion = :tagDescripcion")  
 , @NamedQuery(name = "CatTags.findByQueryInformacion", query = "SELECT c FROM CatTags c WHERE c.queryInformacion = :queryInformacion")  
 , @NamedQuery(name = "CatTags.findByIdVarStatus", query = "SELECT c FROM CatTags c WHERE c.idVarStatus = :idVarStatus")  
 , @NamedQuery(name = "CatTags.findByNombre", query = "SELECT c FROM CatTags c WHERE c.nombre = :nombre")  
 , @NamedQuery(name = "CatTags.findBySistemaId", query = "SELECT c FROM CatTags c WHERE c.sistemaId = :sistemaId")  
 , @NamedQuery(name = "CatTags.findByDescripcion", query = "SELECT c FROM CatTags c WHERE c.descripcion = :descripcion")  
 , @NamedQuery(name = "CatTags.findByModulosAfecta", query = "SELECT c FROM CatTags c WHERE c.modulosAfecta = :modulosAfecta")  
 , @NamedQuery(name = "CatTags.findByActivo", query = "SELECT c FROM CatTags c WHERE c.activo = :activo")  
 , @NamedQuery(name = "CatTags.findByFecheaRegistro", query = "SELECT c FROM CatTags c WHERE c.fecheaRegistro = :fecheaRegistro")  
 , @NamedQuery(name = "CatTags.findByUsuarioRegistroId", query = "SELECT c FROM CatTags c WHERE c.usuarioRegistroId = :usuarioRegistroId")  
 , @NamedQuery(name = "CatTags.findByFechaModificacion", query = "SELECT c FROM CatTags c WHERE c.fechaModificacion = :fechaModificacion")  
 , @NamedQuery(name = "CatTags.findByUsuarioModificoId", query = "SELECT c FROM CatTags c WHERE c.usuarioModificoId = :usuarioModificoId")})  
public class CatTags implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "TAG\_ID")  
 private BigDecimal tagId;  
 @Column(name = "PLANTA\_ID")  
 private BigDecimal plantaId;  
 @Basic(optional = false)  
 @NotNull  
 @Size(min = 1, max = 15)  
 @Column(name = "TAG")  
 private String tag;  
 @Basic(optional = false)  
 @NotNull  
 @Size(min = 1, max = 250)  
 @Column(name = "TAG\_DESCRIPCION")  
 private String tagDescripcion;  
 @Size(max = 1000)  
 @Column(name = "QUERY\_INFORMACION")  
 private String queryInformacion;  
 @Column(name = "ORDEN")  
 private BigInteger orden;  
 @Column(name="ID\_VAR\_STATUS")  
 private BigInteger idVarStatus;  
 @Size(max = 255)  
 @Column(name = "NOMBRE")  
 private String nombre;  
 @Column(name = "SISTEMA\_ID")  
 private BigInteger sistemaId;  
 @Size(max = 4000)  
 @Column(name = "DESCRIPCION")  
 private String descripcion;  
 @Size(max = 250)  
 @Column(name = "INFORMACION\_ADICIONAL")  
 private String informacionAdicional;  
 @Column(name = "INFORMACION\_ADICIONAL\_2")  
 private String informacionAdicional2;  
 @Column(name = "MODULOS\_AFECTA")  
 private BigInteger modulosAfecta;  
 @Column(name = "ACTIVO")  
 private BigInteger activo;  
 @Column(name = "VARIABLE\_ID")  
 private BigInteger variableId;  
 @Column(name = "FECHEA\_REGISTRO")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fecheaRegistro;  
 @Column(name = "USUARIO\_REGISTRO\_ID")  
 private BigInteger usuarioRegistroId;  
 @Column(name = "FECHA\_MODIFICACION")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaModificacion;  
 @Column(name = "USUARIO\_MODIFICO\_ID")  
 private BigInteger usuarioModificoId;  
 @JoinColumn(name = "CLASIFICACION\_QUERY\_ID", referencedColumnName = "CLASIFICACION\_QUERY\_ID")  
 @ManyToOne  
 private CatClasificacionQuery clasificacionQueryId;  
 @OneToMany(mappedBy = "padreId", fetch = FetchType.LAZY)  
 private Set<CatTags> catTagsSet;  
 @JoinColumn(name = "PADRE\_ID", referencedColumnName = "TAG\_ID")  
 @ManyToOne(fetch=FetchType.LAZY)  
 private CatTags padreId;  
  
  
 public void setInformacionAdicional2(String informacionAdicional2) {  
 this.informacionAdicional2 = informacionAdicional2;  
 }  
  
 public String getInformacionAdicional2() {  
 return informacionAdicional2;  
 }  
  
 public CatTags() {  
 }  
  
 public void setIdVarStatus(BigInteger idVarStatus) {  
 this.idVarStatus = idVarStatus;  
 }  
  
 public BigInteger getIdVarStatus() {  
 return idVarStatus;  
 }  
  
 public void setVariableId(BigInteger variableId){  
 this.variableId = variableId;  
 }  
  
 public BigInteger getVariableId(){  
 return variableId;  
 }  
  
 public void setOrden(BigInteger orden) {  
 this.orden = orden;  
 }  
  
 public BigInteger getOrden() {  
 return orden;  
 }  
  
 public CatTags(BigDecimal tagId) {  
 this.tagId = tagId;  
 }  
  
 public CatTags(BigDecimal tagId, String tag, String tagDescripcion) {  
 this.tagId = tagId;  
 this.tag = tag;  
 this.tagDescripcion = tagDescripcion;  
 }  
  
 public BigDecimal getTagId() {  
 return tagId;  
 }  
  
 public void setTagId(BigDecimal tagId) {  
 this.tagId = tagId;  
 }  
  
 public BigDecimal getPlantaId() {  
 return plantaId;  
 }  
  
 public void setPlantaId(BigDecimal plantaId) {  
 this.plantaId = plantaId;  
 }  
  
 public String getTag() {  
 return tag;  
 }  
  
 public void setTag(String tag) {  
 this.tag = tag;  
 }  
  
 public String getTagDescripcion() {  
 return tagDescripcion;  
 }  
  
 public void setTagDescripcion(String tagDescripcion) {  
 this.tagDescripcion = tagDescripcion;  
 }  
  
 public String getQueryInformacion() {  
 return queryInformacion;  
 }  
  
 public void setQueryInformacion(String queryInformacion) {  
 this.queryInformacion = queryInformacion;  
 }  
  
 public String getNombre() {  
 return nombre;  
 }  
  
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
   
  
 public BigInteger getSistemaId() {  
 return sistemaId;  
 }  
  
 public void setSistemaId(BigInteger sistemaId) {  
 this.sistemaId = sistemaId;  
 }  
  
 public String getDescripcion() {  
 return descripcion;  
 }  
  
 public void setDescripcion(String descripcion) {  
 this.descripcion = descripcion;  
 }  
  
 public BigInteger getModulosAfecta() {  
 return modulosAfecta;  
 }  
  
 public void setModulosAfecta(BigInteger modulosAfecta) {  
 this.modulosAfecta = modulosAfecta;  
 }  
  
 public BigInteger getActivo() {  
 return activo;  
 }  
  
 public void setActivo(BigInteger activo) {  
 this.activo = activo;  
 }  
  
 public Date getFecheaRegistro() {  
 return fecheaRegistro;  
 }  
  
 public void setFecheaRegistro(Date fecheaRegistro) {  
 this.fecheaRegistro = fecheaRegistro;  
 }  
  
 public BigInteger getUsuarioRegistroId() {  
 return usuarioRegistroId;  
 }  
  
 public void setUsuarioRegistroId(BigInteger usuarioRegistroId) {  
 this.usuarioRegistroId = usuarioRegistroId;  
 }  
  
 public Date getFechaModificacion() {  
 return fechaModificacion;  
 }  
  
 public void setFechaModificacion(Date fechaModificacion) {  
 this.fechaModificacion = fechaModificacion;  
 }  
  
 public BigInteger getUsuarioModificoId() {  
 return usuarioModificoId;  
 }  
  
 public void setUsuarioModificoId(BigInteger usuarioModificoId) {  
 this.usuarioModificoId = usuarioModificoId;  
 }  
  
 public CatClasificacionQuery getClasificacionQueryId() {  
 return clasificacionQueryId;  
 }  
  
 public void setClasificacionQueryId(CatClasificacionQuery clasificacionQueryId) {  
 this.clasificacionQueryId = clasificacionQueryId;  
 }  
  
 @XmlTransient  
 public Set<CatTags> getCatTagsSet() {  
 return catTagsSet;  
 }  
  
 public void setCatTagsSet(Set<CatTags> catTagsSet) {  
 this.catTagsSet = catTagsSet;  
 }  
  
 public CatTags getPadreId() {  
 return padreId;  
 }  
  
 public void setPadreId(CatTags padreId) {  
 this.padreId = padreId;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (tagId != null ? tagId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatTags)) {  
 return false;  
 }  
 CatTags other = (CatTags) object;  
 if ((this.tagId == null && other.tagId != null) || (this.tagId != null && !this.tagId.equals(other.tagId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatTags[ tagId=" + tagId + " ]";  
 }  
  
 public String getInformacionAdicional() {  
 return informacionAdicional;  
 }  
  
 public void setInformacionAdicional(String informacionAdicional) {  
 this.informacionAdicional = informacionAdicional;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatTipoDato.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
  
import java.math.BigDecimal;  
  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.Table;  
  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
  
import javax.xml.bind.annotation.XmlRootElement;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_TIPO\_DATO")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatTipoDato.findAll", query = "SELECT c FROM CatTipoDato c")  
 , @NamedQuery(name = "CatTipoDato.findByTipoDatoId", query = "SELECT c FROM CatTipoDato c WHERE c.tipoDatoId = :tipoDatoId")  
 , @NamedQuery(name = "CatTipoDato.findByNombre", query = "SELECT c FROM CatTipoDato c WHERE c.nombre = :nombre")})  
public class CatTipoDato implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "TIPO\_DATO\_ID")  
 private BigDecimal tipoDatoId;  
 @Size(max = 50)  
 @Column(name = "NOMBRE")  
 private String nombre;  
  
 public CatTipoDato() {  
 }  
  
 public CatTipoDato(BigDecimal tipoDatoId) {  
 this.tipoDatoId = tipoDatoId;  
 }  
  
 public BigDecimal getTipoDatoId() {  
 return tipoDatoId;  
 }  
  
 public void setTipoDatoId(BigDecimal tipoDatoId) {  
 this.tipoDatoId = tipoDatoId;  
 }  
  
 public String getNombre() {  
 return nombre;  
 }  
  
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (tipoDatoId != null ? tipoDatoId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatTipoDato)) {  
 return false;  
 }  
 CatTipoDato other = (CatTipoDato) object;  
 if ((this.tipoDatoId == null && other.tipoDatoId != null) || (this.tipoDatoId != null && !this.tipoDatoId.equals(other.tipoDatoId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "com.test.entity.CatTipoDato[ tipoDatoId=" + tipoDatoId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatTipoDocumento.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
  
import java.math.BigDecimal;  
  
import java.util.Set;  
  
import javax.persistence.Basic;  
import javax.persistence.CascadeType;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlTransient;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_TIPO\_DOCUMENTO")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatTipoDocumento.findAll", query = "SELECT c FROM CatTipoDocumento c")  
 , @NamedQuery(name = "CatTipoDocumento.findByTipoDocumentoId", query = "SELECT c FROM CatTipoDocumento c WHERE c.tipoDocumentoId = :tipoDocumentoId")  
 , @NamedQuery(name = "CatTipoDocumento.findByTipo", query = "SELECT c FROM CatTipoDocumento c WHERE c.tipo = :tipo")  
 , @NamedQuery(name = "CatTipoDocumento.findByExtension", query = "SELECT c FROM CatTipoDocumento c WHERE c.extension = :extension")})  
public class CatTipoDocumento implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "TIPO\_DOCUMENTO\_ID")  
 private BigDecimal tipoDocumentoId;  
 @Basic(optional = false)  
 @NotNull  
 @Size(min = 1, max = 100)  
 @Column(name = "TIPO")  
 private String tipo;  
 @Basic(optional = false)  
 @NotNull  
 @Size(min = 1, max = 15)  
 @Column(name = "EXTENSION")  
 private String extension;  
 @OneToMany(cascade = CascadeType.ALL, mappedBy = "tipoDocumentoId")  
 private Set<TxAdjuntos> txAdjuntosSet;  
  
 public CatTipoDocumento() {  
 }  
  
 public CatTipoDocumento(BigDecimal tipoDocumentoId) {  
 this.tipoDocumentoId = tipoDocumentoId;  
 }  
  
 public CatTipoDocumento(BigDecimal tipoDocumentoId, String tipo, String extension) {  
 this.tipoDocumentoId = tipoDocumentoId;  
 this.tipo = tipo;  
 this.extension = extension;  
 }  
  
 public BigDecimal getTipoDocumentoId() {  
 return tipoDocumentoId;  
 }  
  
 public void setTipoDocumentoId(BigDecimal tipoDocumentoId) {  
 this.tipoDocumentoId = tipoDocumentoId;  
 }  
  
 public String getTipo() {  
 return tipo;  
 }  
  
 public void setTipo(String tipo) {  
 this.tipo = tipo;  
 }  
  
 public String getExtension() {  
 return extension;  
 }  
  
 public void setExtension(String extension) {  
 this.extension = extension;  
 }  
  
 @XmlTransient  
 public Set<TxAdjuntos> getTxAdjuntosSet() {  
 return txAdjuntosSet;  
 }  
  
 public void setTxAdjuntosSet(Set<TxAdjuntos> txAdjuntosSet) {  
 this.txAdjuntosSet = txAdjuntosSet;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (tipoDocumentoId != null ? tipoDocumentoId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatTipoDocumento)) {  
 return false;  
 }  
 CatTipoDocumento other = (CatTipoDocumento) object;  
 if ((this.tipoDocumentoId == null && other.tipoDocumentoId != null) || (this.tipoDocumentoId != null && !this.tipoDocumentoId.equals(other.tipoDocumentoId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatTipoDocumento[ tipoDocumentoId=" + tipoDocumentoId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatTipoFactoresSat.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatTipoFactoresSat generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_TIPO\_FACTORES\_SAT"  
)  
public class CatTipoFactoresSat implements java.io.Serializable {  
  
  
 private BigDecimal tipoFactorSatId;  
 private String nombre;  
 private Boolean activo;  
 private String claveSat;  
 private Set<CatTipoImpuestos> catTipoImpuestoses = new HashSet<CatTipoImpuestos>(0);  
  
 public CatTipoFactoresSat() {  
 }  
  
 public CatTipoFactoresSat(String nombre, Boolean activo, String claveSat, Set<CatTipoImpuestos> catTipoImpuestoses) {  
 this.nombre = nombre;  
 this.activo = activo;  
 this.claveSat = claveSat;  
 this.catTipoImpuestoses = catTipoImpuestoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="TIPO\_FACTOR\_SAT\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getTipoFactorSatId() {  
 return this.tipoFactorSatId;  
 }  
   
 public void setTipoFactorSatId(BigDecimal tipoFactorSatId) {  
 this.tipoFactorSatId = tipoFactorSatId;  
 }  
  
   
 @Column(name="NOMBRE")  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="ACTIVO", precision=1, scale=0)  
 public Boolean getActivo() {  
 return this.activo;  
 }  
   
 public void setActivo(Boolean activo) {  
 this.activo = activo;  
 }  
  
   
 @Column(name="CLAVE\_SAT", length=10)  
 public String getClaveSat() {  
 return this.claveSat;  
 }  
   
 public void setClaveSat(String claveSat) {  
 this.claveSat = claveSat;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catTipoFactoresSat")  
 public Set<CatTipoImpuestos> getCatTipoImpuestoses() {  
 return this.catTipoImpuestoses;  
 }  
   
 public void setCatTipoImpuestoses(Set<CatTipoImpuestos> catTipoImpuestoses) {  
 this.catTipoImpuestoses = catTipoImpuestoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatTipoImpuestos.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import java.util.HashSet;  
import java.util.Set;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatTipoImpuestos generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_TIPO\_IMPUESTOS"  
)  
public class CatTipoImpuestos implements java.io.Serializable {  
  
  
 private BigDecimal tipoImpuestoId;  
 private CatTipoFactoresSat catTipoFactoresSat;  
 private String nombre;  
 private String claveSat;  
 private Boolean retenido;  
 private Boolean trasladado;  
 private Boolean federal;  
 private Boolean exento;  
 private Boolean activo;  
 private Set<CatImpuestos> catImpuestoses = new HashSet<CatImpuestos>(0);  
  
 public CatTipoImpuestos() {  
 }  
  
 public CatTipoImpuestos(CatTipoFactoresSat catTipoFactoresSat, String nombre, String claveSat, Boolean retenido, Boolean trasladado, Boolean federal, Boolean exento, Boolean activo, Set<CatImpuestos> catImpuestoses) {  
 this.catTipoFactoresSat = catTipoFactoresSat;  
 this.nombre = nombre;  
 this.claveSat = claveSat;  
 this.retenido = retenido;  
 this.trasladado = trasladado;  
 this.federal = federal;  
 this.exento = exento;  
 this.activo = activo;  
 this.catImpuestoses = catImpuestoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="TIPO\_IMPUESTO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getTipoImpuestoId() {  
 return this.tipoImpuestoId;  
 }  
   
 public void setTipoImpuestoId(BigDecimal tipoImpuestoId) {  
 this.tipoImpuestoId = tipoImpuestoId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="TIPO\_FACTOR\_SAT\_ID")  
 public CatTipoFactoresSat getCatTipoFactoresSat() {  
 return this.catTipoFactoresSat;  
 }  
   
 public void setCatTipoFactoresSat(CatTipoFactoresSat catTipoFactoresSat) {  
 this.catTipoFactoresSat = catTipoFactoresSat;  
 }  
  
   
 @Column(name="NOMBRE", length=10)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="CLAVE\_SAT", length=2)  
 public String getClaveSat() {  
 return this.claveSat;  
 }  
   
 public void setClaveSat(String claveSat) {  
 this.claveSat = claveSat;  
 }  
  
   
 @Column(name="RETENIDO", precision=1, scale=0)  
 public Boolean getRetenido() {  
 return this.retenido;  
 }  
   
 public void setRetenido(Boolean retenido) {  
 this.retenido = retenido;  
 }  
  
   
 @Column(name="TRASLADADO", precision=1, scale=0)  
 public Boolean getTrasladado() {  
 return this.trasladado;  
 }  
   
 public void setTrasladado(Boolean trasladado) {  
 this.trasladado = trasladado;  
 }  
  
   
 @Column(name="FEDERAL", precision=1, scale=0)  
 public Boolean getFederal() {  
 return this.federal;  
 }  
   
 public void setFederal(Boolean federal) {  
 this.federal = federal;  
 }  
  
   
 @Column(name="EXENTO", precision=1, scale=0)  
 public Boolean getExento() {  
 return this.exento;  
 }  
   
 public void setExento(Boolean exento) {  
 this.exento = exento;  
 }  
  
   
 @Column(name="ACTIVO", precision=1, scale=0)  
 public Boolean getActivo() {  
 return this.activo;  
 }  
   
 public void setActivo(Boolean activo) {  
 this.activo = activo;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catTipoImpuestos")  
 public Set<CatImpuestos> getCatImpuestoses() {  
 return this.catImpuestoses;  
 }  
   
 public void setCatImpuestoses(Set<CatImpuestos> catImpuestoses) {  
 this.catImpuestoses = catImpuestoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatTipoRelacion.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatTipoRelacion generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_TIPO\_RELACION"  
)  
public class CatTipoRelacion implements java.io.Serializable {  
  
  
 private BigDecimal tipoRelacionId;  
 private String nombre;  
 private Set<TxFacturaUuids> txFacturaUuidses = new HashSet<TxFacturaUuids>(0);  
 private Set<TxDocumentos> txDocumentoses = new HashSet<TxDocumentos>(0);  
  
 public CatTipoRelacion() {  
 }  
  
 public CatTipoRelacion(String nombre, Set<TxFacturaUuids> txFacturaUuidses, Set<TxDocumentos> txDocumentoses) {  
 this.nombre = nombre;  
 this.txFacturaUuidses = txFacturaUuidses;  
 this.txDocumentoses = txDocumentoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="TIPO\_RELACION\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getTipoRelacionId() {  
 return this.tipoRelacionId;  
 }  
   
 public void setTipoRelacionId(BigDecimal tipoRelacionId) {  
 this.tipoRelacionId = tipoRelacionId;  
 }  
  
   
 @Column(name="NOMBRE")  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catTipoRelacion")  
 public Set<TxFacturaUuids> getTxFacturaUuidses() {  
 return this.txFacturaUuidses;  
 }  
   
 public void setTxFacturaUuidses(Set<TxFacturaUuids> txFacturaUuidses) {  
 this.txFacturaUuidses = txFacturaUuidses;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catTipoRelacion")  
 public Set<TxDocumentos> getTxDocumentoses() {  
 return this.txDocumentoses;  
 }  
   
 public void setTxDocumentoses(Set<TxDocumentos> txDocumentoses) {  
 this.txDocumentoses = txDocumentoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatTiposDeEvento.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import java.math.BigInteger;  
import java.util.Date;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
import javax.xml.bind.annotation.XmlRootElement;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_TIPOS\_DE\_EVENTO")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatTiposDeEvento.findAll", query = "SELECT c FROM CatTiposDeEvento c")  
 , @NamedQuery(name = "CatTiposDeEvento.findByTiposDeEventoId", query = "SELECT c FROM CatTiposDeEvento c WHERE c.tiposDeEventoId = :tiposDeEventoId")  
 , @NamedQuery(name = "CatTiposDeEvento.findByNombre", query = "SELECT c FROM CatTiposDeEvento c WHERE c.nombre = :nombre")  
 , @NamedQuery(name = "CatTiposDeEvento.findBySiglas", query = "SELECT c FROM CatTiposDeEvento c WHERE c.siglas = :siglas")  
 , @NamedQuery(name = "CatTiposDeEvento.findByVisibleEnBitacora", query = "SELECT c FROM CatTiposDeEvento c WHERE c.visibleEnBitacora = :visibleEnBitacora")  
 , @NamedQuery(name = "CatTiposDeEvento.findByDescripcion", query = "SELECT c FROM CatTiposDeEvento c WHERE c.descripcion = :descripcion")  
 , @NamedQuery(name = "CatTiposDeEvento.findByActivo", query = "SELECT c FROM CatTiposDeEvento c WHERE c.activo = :activo")  
 , @NamedQuery(name = "CatTiposDeEvento.findByFechaRegistro", query = "SELECT c FROM CatTiposDeEvento c WHERE c.fechaRegistro = :fechaRegistro")  
 , @NamedQuery(name = "CatTiposDeEvento.findByUsuarioRegistro", query = "SELECT c FROM CatTiposDeEvento c WHERE c.usuarioRegistro = :usuarioRegistro")  
 , @NamedQuery(name = "CatTiposDeEvento.findByFechaModificacion", query = "SELECT c FROM CatTiposDeEvento c WHERE c.fechaModificacion = :fechaModificacion")  
 , @NamedQuery(name = "CatTiposDeEvento.findByUsuarioModificoId", query = "SELECT c FROM CatTiposDeEvento c WHERE c.usuarioModificoId = :usuarioModificoId")})  
public class CatTiposDeEvento implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "TIPOS\_DE\_EVENTO\_ID")  
 private BigDecimal tiposDeEventoId;  
 @Size(max = 1020)  
 @Column(name = "NOMBRE")  
 private String nombre;  
 @Size(max = 4000)  
 @Column(name = "SIGLAS")  
 private String siglas;  
 @Column(name = "VISIBLE\_EN\_BITACORA")  
 private BigInteger visibleEnBitacora;  
 @Size(max = 4000)  
 @Column(name = "DESCRIPCION")  
 private String descripcion;  
 @Column(name = "ACTIVO")  
 private BigInteger activo;  
 @Column(name = "FECHA\_REGISTRO")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaRegistro;  
 @Column(name = "USUARIO\_REGISTRO")  
 private BigInteger usuarioRegistro;  
 @Column(name = "FECHA\_MODIFICACION")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaModificacion;  
 @Column(name = "USUARIO\_MODIFICO\_ID")  
 private BigInteger usuarioModificoId;  
  
 public CatTiposDeEvento() {  
 }  
  
 public CatTiposDeEvento(BigDecimal tiposDeEventoId) {  
 this.tiposDeEventoId = tiposDeEventoId;  
 }  
  
 public BigDecimal getTiposDeEventoId() {  
 return tiposDeEventoId;  
 }  
  
 public void setTiposDeEventoId(BigDecimal tiposDeEventoId) {  
 this.tiposDeEventoId = tiposDeEventoId;  
 }  
  
 public String getNombre() {  
 return nombre;  
 }  
  
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
 public String getSiglas() {  
 return siglas;  
 }  
  
 public void setSiglas(String siglas) {  
 this.siglas = siglas;  
 }  
  
 public BigInteger getVisibleEnBitacora() {  
 return visibleEnBitacora;  
 }  
  
 public void setVisibleEnBitacora(BigInteger visibleEnBitacora) {  
 this.visibleEnBitacora = visibleEnBitacora;  
 }  
  
 public String getDescripcion() {  
 return descripcion;  
 }  
  
 public void setDescripcion(String descripcion) {  
 this.descripcion = descripcion;  
 }  
  
 public BigInteger getActivo() {  
 return activo;  
 }  
  
 public void setActivo(BigInteger activo) {  
 this.activo = activo;  
 }  
  
 public Date getFechaRegistro() {  
 return fechaRegistro;  
 }  
  
 public void setFechaRegistro(Date fechaRegistro) {  
 this.fechaRegistro = fechaRegistro;  
 }  
  
 public BigInteger getUsuarioRegistro() {  
 return usuarioRegistro;  
 }  
  
 public void setUsuarioRegistro(BigInteger usuarioRegistro) {  
 this.usuarioRegistro = usuarioRegistro;  
 }  
  
 public Date getFechaModificacion() {  
 return fechaModificacion;  
 }  
  
 public void setFechaModificacion(Date fechaModificacion) {  
 this.fechaModificacion = fechaModificacion;  
 }  
  
 public BigInteger getUsuarioModificoId() {  
 return usuarioModificoId;  
 }  
  
 public void setUsuarioModificoId(BigInteger usuarioModificoId) {  
 this.usuarioModificoId = usuarioModificoId;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (tiposDeEventoId != null ? tiposDeEventoId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatTiposDeEvento)) {  
 return false;  
 }  
 CatTiposDeEvento other = (CatTiposDeEvento) object;  
 if ((this.tiposDeEventoId == null && other.tiposDeEventoId != null) || (this.tiposDeEventoId != null && !this.tiposDeEventoId.equals(other.tiposDeEventoId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "com.test.entity.CatTiposDeEvento[ tiposDeEventoId=" + tiposDeEventoId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatTiposFacturas.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatTiposFacturas generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_TIPOS\_FACTURAS"  
)  
public class CatTiposFacturas implements java.io.Serializable {  
  
  
 private BigDecimal tipoFacturaId;  
 private String nombre;  
 private String serie;  
 private Set<CatPlantaTipoFacturas> catPlantaTipoFacturases = new HashSet<CatPlantaTipoFacturas>(0);  
 private Set<TxDocumentos> txDocumentoses = new HashSet<TxDocumentos>(0);  
  
 public CatTiposFacturas() {  
 }  
  
 public CatTiposFacturas(String nombre, String serie, Set<CatPlantaTipoFacturas> catPlantaTipoFacturases, Set<TxDocumentos> txDocumentoses) {  
 this.nombre = nombre;  
 this.serie = serie;  
 this.catPlantaTipoFacturases = catPlantaTipoFacturases;  
 this.txDocumentoses = txDocumentoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="TIPO\_FACTURA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getTipoFacturaId() {  
 return this.tipoFacturaId;  
 }  
   
 public void setTipoFacturaId(BigDecimal tipoFacturaId) {  
 this.tipoFacturaId = tipoFacturaId;  
 }  
  
   
 @Column(name="NOMBRE")  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="SERIE", length=3)  
 public String getSerie() {  
 return this.serie;  
 }  
   
 public void setSerie(String serie) {  
 this.serie = serie;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catTiposFacturas")  
 public Set<CatPlantaTipoFacturas> getCatPlantaTipoFacturases() {  
 return this.catPlantaTipoFacturases;  
 }  
   
 public void setCatPlantaTipoFacturases(Set<CatPlantaTipoFacturas> catPlantaTipoFacturases) {  
 this.catPlantaTipoFacturases = catPlantaTipoFacturases;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catTiposFacturas")  
 public Set<TxDocumentos> getTxDocumentoses() {  
 return this.txDocumentoses;  
 }  
   
 public void setTxDocumentoses(Set<TxDocumentos> txDocumentoses) {  
 this.txDocumentoses = txDocumentoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatTiposServicios.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatTiposServicios generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_TIPOS\_SERVICIOS"  
)  
public class CatTiposServicios implements java.io.Serializable {  
  
  
 private BigDecimal tipoServicioId;  
 private String nombre;  
 private Set<CatServicios> catServicioses = new HashSet<CatServicios>(0);  
  
 public CatTiposServicios() {  
 }  
  
 public CatTiposServicios(String nombre, Set<CatServicios> catServicioses) {  
 this.nombre = nombre;  
 this.catServicioses = catServicioses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="TIPO\_SERVICIO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getTipoServicioId() {  
 return this.tipoServicioId;  
 }  
   
 public void setTipoServicioId(BigDecimal tipoServicioId) {  
 this.tipoServicioId = tipoServicioId;  
 }  
  
   
 @Column(name="NOMBRE")  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catTiposServicios")  
 public Set<CatServicios> getCatServicioses() {  
 return this.catServicioses;  
 }  
   
 public void setCatServicioses(Set<CatServicios> catServicioses) {  
 this.catServicioses = catServicioses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatUnidad.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatUnidad generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_UNIDAD"  
)  
public class CatUnidad implements java.io.Serializable {  
  
  
 private BigDecimal unidadId;  
 private String unidad;  
  
 public CatUnidad() {  
 }  
  
 public CatUnidad(String unidad) {  
 this.unidad = unidad;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="UNIDAD\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getUnidadId() {  
 return this.unidadId;  
 }  
   
 public void setUnidadId(BigDecimal unidadId) {  
 this.unidadId = unidadId;  
 }  
  
   
 @Column(name="UNIDAD", nullable=false, length=150)  
 public String getUnidad() {  
 return this.unidad;  
 }  
   
 public void setUnidad(String unidad) {  
 this.unidad = unidad;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatUnidades.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import java.math.BigInteger;  
import java.util.Date;  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
import javax.xml.bind.annotation.XmlRootElement;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "CAT\_UNIDADES")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "CatUnidades.findAll", query = "SELECT c FROM CatUnidades c")  
 , @NamedQuery(name = "CatUnidades.findByUnidadId", query = "SELECT c FROM CatUnidades c WHERE c.unidadId = :unidadId")  
 , @NamedQuery(name = "CatUnidades.findByNombre", query = "SELECT c FROM CatUnidades c WHERE c.nombre = :nombre")  
 , @NamedQuery(name = "CatUnidades.findBySiglas", query = "SELECT c FROM CatUnidades c WHERE c.siglas = :siglas")  
 , @NamedQuery(name = "CatUnidades.findByVisibleEnBitacora", query = "SELECT c FROM CatUnidades c WHERE c.visibleEnBitacora = :visibleEnBitacora")  
 , @NamedQuery(name = "CatUnidades.findByDescripcion", query = "SELECT c FROM CatUnidades c WHERE c.descripcion = :descripcion")  
 , @NamedQuery(name = "CatUnidades.findByActivo", query = "SELECT c FROM CatUnidades c WHERE c.activo = :activo")  
 , @NamedQuery(name = "CatUnidades.findByFechaRegistro", query = "SELECT c FROM CatUnidades c WHERE c.fechaRegistro = :fechaRegistro")  
 , @NamedQuery(name = "CatUnidades.findByUsuarioRegistro", query = "SELECT c FROM CatUnidades c WHERE c.usuarioRegistro = :usuarioRegistro")  
 , @NamedQuery(name = "CatUnidades.findByFechaModificacion", query = "SELECT c FROM CatUnidades c WHERE c.fechaModificacion = :fechaModificacion")  
 , @NamedQuery(name = "CatUnidades.findByUsuarioModificoId", query = "SELECT c FROM CatUnidades c WHERE c.usuarioModificoId = :usuarioModificoId")})  
public class CatUnidades implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "UNIDAD\_ID")  
 private BigDecimal unidadId;  
 @Size(max = 1020)  
 @Column(name = "NOMBRE")  
 private String nombre;  
 @Size(max = 4000)  
 @Column(name = "SIGLAS")  
 private String siglas;  
 @Column(name = "VISIBLE\_EN\_BITACORA")  
 private BigInteger visibleEnBitacora;  
 @Size(max = 4000)  
 @Column(name = "DESCRIPCION")  
 private String descripcion;  
 @Column(name = "ACTIVO")  
 private BigInteger activo;  
 @Column(name = "FECHA\_REGISTRO")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaRegistro;  
 @Column(name = "USUARIO\_REGISTRO")  
 private BigInteger usuarioRegistro;  
 @Column(name = "FECHA\_MODIFICACION")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaModificacion;  
 @Column(name = "USUARIO\_MODIFICO\_ID")  
 private BigInteger usuarioModificoId;  
  
 public CatUnidades() {  
 }  
  
 public CatUnidades(BigDecimal unidadId) {  
 this.unidadId = unidadId;  
 }  
  
 public BigDecimal getUnidadId() {  
 return unidadId;  
 }  
  
 public void setUnidadId(BigDecimal unidadId) {  
 this.unidadId = unidadId;  
 }  
  
 public String getNombre() {  
 return nombre;  
 }  
  
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
 public String getSiglas() {  
 return siglas;  
 }  
  
 public void setSiglas(String siglas) {  
 this.siglas = siglas;  
 }  
  
 public BigInteger getVisibleEnBitacora() {  
 return visibleEnBitacora;  
 }  
  
 public void setVisibleEnBitacora(BigInteger visibleEnBitacora) {  
 this.visibleEnBitacora = visibleEnBitacora;  
 }  
  
 public String getDescripcion() {  
 return descripcion;  
 }  
  
 public void setDescripcion(String descripcion) {  
 this.descripcion = descripcion;  
 }  
  
 public BigInteger getActivo() {  
 return activo;  
 }  
  
 public void setActivo(BigInteger activo) {  
 this.activo = activo;  
 }  
  
 public Date getFechaRegistro() {  
 return fechaRegistro;  
 }  
  
 public void setFechaRegistro(Date fechaRegistro) {  
 this.fechaRegistro = fechaRegistro;  
 }  
  
 public BigInteger getUsuarioRegistro() {  
 return usuarioRegistro;  
 }  
  
 public void setUsuarioRegistro(BigInteger usuarioRegistro) {  
 this.usuarioRegistro = usuarioRegistro;  
 }  
  
 public Date getFechaModificacion() {  
 return fechaModificacion;  
 }  
  
 public void setFechaModificacion(Date fechaModificacion) {  
 this.fechaModificacion = fechaModificacion;  
 }  
  
 public BigInteger getUsuarioModificoId() {  
 return usuarioModificoId;  
 }  
  
 public void setUsuarioModificoId(BigInteger usuarioModificoId) {  
 this.usuarioModificoId = usuarioModificoId;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (unidadId != null ? unidadId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof CatUnidades)) {  
 return false;  
 }  
 CatUnidades other = (CatUnidades) object;  
 if ((this.unidadId == null && other.unidadId != null) || (this.unidadId != null && !this.unidadId.equals(other.unidadId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.CatUnidades[ unidadId=" + unidadId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatUsoCfdi.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import java.util.HashSet;  
import java.util.Set;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatUsoCfdi generated by hbm2java  
 \*/  
@Entity  
@Table(name = "CAT\_USO\_CFDI")  
public class CatUsoCfdi implements java.io.Serializable {  
  
  
 private BigDecimal usoCfdiId;  
 private String nombre;  
 private String clave;  
 private int paraPersonaFisica;  
 private int paraPersonaMoral;  
 private int activo;  
 private Set<TxDocumentos> txDocumentoses = new HashSet<TxDocumentos>(0);  
  
 public CatUsoCfdi() {  
 }  
  
 public CatUsoCfdi(String usoCfdi, Set<TxDocumentos> txDocumentoses) {  
 this.nombre = usoCfdi;  
 this.txDocumentoses = txDocumentoses;  
 }  
  
 @Id  
 @GeneratedValue(strategy = IDENTITY)  
  
  
 @Column(name = "USO\_CFDI\_ID", unique = true, nullable = false, precision = 22, scale = 0)  
 public BigDecimal getUsoCfdiId() {  
 return this.usoCfdiId;  
 }  
  
 public void setUsoCfdiId(BigDecimal usoCfdiId) {  
 this.usoCfdiId = usoCfdiId;  
 }  
  
  
 @Column(name = "NOMBRE")  
 public String getNombre() {  
 return this.nombre;  
 }  
  
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "catUsoCfdi")  
 public Set<TxDocumentos> getTxDocumentoses() {  
 return this.txDocumentoses;  
 }  
  
 public void setTxDocumentoses(Set<TxDocumentos> txDocumentoses) {  
 this.txDocumentoses = txDocumentoses;  
 }  
  
 public void setClave(String clave) {  
 this.clave = clave;  
 }  
  
 @Column(name = "CLAVE")  
 public String getClave() {  
 return clave;  
 }  
  
 public void setParaPersonaFisica(int paraPersonaFisica) {  
 this.paraPersonaFisica = paraPersonaFisica;  
 }  
  
 @Column(name = "PARA\_PERSONA\_FISICA")  
 public int getParaPersonaFisica() {  
 return paraPersonaFisica;  
 }  
  
 public void setParaPersonaMoral(int paraPersonaMoral) {  
 this.paraPersonaMoral = paraPersonaMoral;  
 }  
  
 @Column(name = "PARA\_PERSONA\_MORAL")  
 public int getParaPersonaMoral() {  
 return paraPersonaMoral;  
 }  
  
 public void setActivo(int activo) {  
 this.activo = activo;  
 }  
  
 @Column(name = "ACTIVO")  
 public int getActivo() {  
 return activo;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatVariable.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatVariable generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_VARIABLE"  
)  
public class CatVariable implements java.io.Serializable {  
  
  
 private BigDecimal variableId;  
 private CatPlanta catPlanta;  
 private String unidadMedida;  
 private String nombre;  
 private String tipo;  
 private BigDecimal valorMinimo;  
 private BigDecimal valorMaximo;  
 private BigDecimal diferenciaMaxima;  
 private Set<CatReglaNegocio> catReglaNegocios = new HashSet<CatReglaNegocio>(0);  
 private Set<CatLimitesDominio> catLimitesDominios = new HashSet<CatLimitesDominio>(0);  
 private Set<CatRangoVariable> catRangoVariables = new HashSet<CatRangoVariable>(0);  
 private Set<CatCurvasCorreccion> catCurvasCorreccions = new HashSet<CatCurvasCorreccion>(0);  
 private Set<TxCincominutalVar> txCincominutalVars = new HashSet<TxCincominutalVar>(0);  
  
 public CatVariable() {  
 }  
  
   
 public CatVariable(String unidadMedida, String nombre, String tipo) {  
 this.unidadMedida = unidadMedida;  
 this.nombre = nombre;  
 this.tipo = tipo;  
 }  
 public CatVariable(CatPlanta catPlanta, String unidadMedida, String nombre, String tipo, BigDecimal valorMinimo, BigDecimal valorMaximo, BigDecimal diferenciaMaxima, Set<CatReglaNegocio> catReglaNegocios, Set<CatLimitesDominio> catLimitesDominios, Set<CatRangoVariable> catRangoVariables, Set<CatCurvasCorreccion> catCurvasCorreccions, Set<TxCincominutalVar> txCincominutalVars) {  
 this.catPlanta = catPlanta;  
 this.unidadMedida = unidadMedida;  
 this.nombre = nombre;  
 this.tipo = tipo;  
 this.valorMinimo = valorMinimo;  
 this.valorMaximo = valorMaximo;  
 this.diferenciaMaxima = diferenciaMaxima;  
 this.catReglaNegocios = catReglaNegocios;  
 this.catLimitesDominios = catLimitesDominios;  
 this.catRangoVariables = catRangoVariables;  
 this.catCurvasCorreccions = catCurvasCorreccions;  
 this.txCincominutalVars = txCincominutalVars;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="VARIABLE\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getVariableId() {  
 return this.variableId;  
 }  
   
 public void setVariableId(BigDecimal variableId) {  
 this.variableId = variableId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="PLANTA\_ID")  
 public CatPlanta getCatPlanta() {  
 return this.catPlanta;  
 }  
   
 public void setCatPlanta(CatPlanta catPlanta) {  
 this.catPlanta = catPlanta;  
 }  
  
   
 @Column(name="UNIDAD\_MEDIDA", nullable=false, length=50)  
 public String getUnidadMedida() {  
 return this.unidadMedida;  
 }  
   
 public void setUnidadMedida(String unidadMedida) {  
 this.unidadMedida = unidadMedida;  
 }  
  
   
 @Column(name="NOMBRE", nullable=false, length=100)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="TIPO", nullable=false, length=1)  
 public String getTipo() {  
 return this.tipo;  
 }  
   
 public void setTipo(String tipo) {  
 this.tipo = tipo;  
 }  
  
   
 @Column(name="VALOR\_MINIMO", precision=22, scale=0)  
 public BigDecimal getValorMinimo() {  
 return this.valorMinimo;  
 }  
   
 public void setValorMinimo(BigDecimal valorMinimo) {  
 this.valorMinimo = valorMinimo;  
 }  
  
   
 @Column(name="VALOR\_MAXIMO", precision=22, scale=0)  
 public BigDecimal getValorMaximo() {  
 return this.valorMaximo;  
 }  
   
 public void setValorMaximo(BigDecimal valorMaximo) {  
 this.valorMaximo = valorMaximo;  
 }  
  
   
 @Column(name="DIFERENCIA\_MAXIMA", precision=22, scale=0)  
 public BigDecimal getDiferenciaMaxima() {  
 return this.diferenciaMaxima;  
 }  
   
 public void setDiferenciaMaxima(BigDecimal diferenciaMaxima) {  
 this.diferenciaMaxima = diferenciaMaxima;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catVariable")  
 public Set<CatReglaNegocio> getCatReglaNegocios() {  
 return this.catReglaNegocios;  
 }  
   
 public void setCatReglaNegocios(Set<CatReglaNegocio> catReglaNegocios) {  
 this.catReglaNegocios = catReglaNegocios;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catVariable")  
 public Set<CatLimitesDominio> getCatLimitesDominios() {  
 return this.catLimitesDominios;  
 }  
   
 public void setCatLimitesDominios(Set<CatLimitesDominio> catLimitesDominios) {  
 this.catLimitesDominios = catLimitesDominios;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catVariable")  
 public Set<CatRangoVariable> getCatRangoVariables() {  
 return this.catRangoVariables;  
 }  
   
 public void setCatRangoVariables(Set<CatRangoVariable> catRangoVariables) {  
 this.catRangoVariables = catRangoVariables;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catVariable")  
 public Set<CatCurvasCorreccion> getCatCurvasCorreccions() {  
 return this.catCurvasCorreccions;  
 }  
   
 public void setCatCurvasCorreccions(Set<CatCurvasCorreccion> catCurvasCorreccions) {  
 this.catCurvasCorreccions = catCurvasCorreccions;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catVariable")  
 public Set<TxCincominutalVar> getTxCincominutalVars() {  
 return this.txCincominutalVars;  
 }  
   
 public void setTxCincominutalVars(Set<TxCincominutalVar> txCincominutalVars) {  
 this.txCincominutalVars = txCincominutalVars;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatVariableFacturacion.java

package mx.mbi.dgc.jpa.entity;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
  
@Entity  
@Table(name = "CAT\_VARIABLE\_FACTURACION")  
public class CatVariableFacturacion {  
 private Integer variableFacturacionId;  
 private String nombre;  
  
 public CatVariableFacturacion() {  
  
 }  
  
 @Id  
 @GeneratedValue(strategy = IDENTITY)  
  
 @Column(name = "VARIABLE\_FACTURACION\_ID", nullable = false, unique = true)  
 public Integer getVariableFacturacionId() {  
 return variableFacturacionId;  
 }  
  
 public void setVariableFacturacionId(Integer variableFacturacionId) {  
 this.variableFacturacionId = variableFacturacionId;  
 }  
  
 @Column(name = "NOMBRE")  
 public String getNombre() {  
 return nombre;  
 }  
  
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatVariablePerfil.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatVariablePerfil generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_VARIABLE\_PERFIL"  
)  
public class CatVariablePerfil implements java.io.Serializable {  
  
  
 private BigDecimal variablePerfilId;  
 private CatPlanta catPlanta;  
 private String variablePerfil;  
 private String unidadPerfil;  
 private BigDecimal limiteSuperior;  
 private BigDecimal limiteInferior;  
 private Set<TxPerfilesVariable> txPerfilesVariables = new HashSet<TxPerfilesVariable>(0);  
  
 public CatVariablePerfil() {  
 }  
  
 public CatVariablePerfil(CatPlanta catPlanta, String variablePerfil, String unidadPerfil, BigDecimal limiteSuperior, BigDecimal limiteInferior, Set<TxPerfilesVariable> txPerfilesVariables) {  
 this.catPlanta = catPlanta;  
 this.variablePerfil = variablePerfil;  
 this.unidadPerfil = unidadPerfil;  
 this.limiteSuperior = limiteSuperior;  
 this.limiteInferior = limiteInferior;  
 this.txPerfilesVariables = txPerfilesVariables;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="VARIABLE\_PERFIL\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getVariablePerfilId() {  
 return this.variablePerfilId;  
 }  
   
 public void setVariablePerfilId(BigDecimal variablePerfilId) {  
 this.variablePerfilId = variablePerfilId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="PLANTA\_ID")  
 public CatPlanta getCatPlanta() {  
 return this.catPlanta;  
 }  
   
 public void setCatPlanta(CatPlanta catPlanta) {  
 this.catPlanta = catPlanta;  
 }  
  
   
 @Column(name="VARIABLE\_PERFIL", length=150)  
 public String getVariablePerfil() {  
 return this.variablePerfil;  
 }  
   
 public void setVariablePerfil(String variablePerfil) {  
 this.variablePerfil = variablePerfil;  
 }  
  
   
 @Column(name="UNIDAD\_PERFIL", length=150)  
 public String getUnidadPerfil() {  
 return this.unidadPerfil;  
 }  
   
 public void setUnidadPerfil(String unidadPerfil) {  
 this.unidadPerfil = unidadPerfil;  
 }  
  
   
 @Column(name="LIMITE\_SUPERIOR", precision=22, scale=0)  
 public BigDecimal getLimiteSuperior() {  
 return this.limiteSuperior;  
 }  
   
 public void setLimiteSuperior(BigDecimal limiteSuperior) {  
 this.limiteSuperior = limiteSuperior;  
 }  
  
   
 @Column(name="LIMITE\_INFERIOR", precision=22, scale=0)  
 public BigDecimal getLimiteInferior() {  
 return this.limiteInferior;  
 }  
   
 public void setLimiteInferior(BigDecimal limiteInferior) {  
 this.limiteInferior = limiteInferior;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catVariablePerfil")  
 public Set<TxPerfilesVariable> getTxPerfilesVariables() {  
 return this.txPerfilesVariables;  
 }  
   
 public void setTxPerfilesVariables(Set<TxPerfilesVariable> txPerfilesVariables) {  
 this.txPerfilesVariables = txPerfilesVariables;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\CatVersion.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* CatVersion generated by hbm2java  
 \*/  
@Entity  
@Table(name="CAT\_VERSION"  
)  
public class CatVersion implements java.io.Serializable {  
  
  
 private BigDecimal versionId;  
 private String version;  
 private Set<TxDocumentos> txDocumentoses = new HashSet<TxDocumentos>(0);  
  
 public CatVersion() {  
 }  
  
 public CatVersion(Set<TxDocumentos> txDocumentoses) {  
 this.txDocumentoses = txDocumentoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="VERSION\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getVersionId() {  
 return this.versionId;  
 }  
   
 public void setVersionId(BigDecimal versionId) {  
 this.versionId = versionId;  
 }  
  
 @Column(name="VERSION", length=10)  
 public String getVersion() {  
 return this.version;  
 }  
   
 public void setVersion(String version) {  
 this.version = version;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="catVersion")  
 public Set<TxDocumentos> getTxDocumentoses() {  
 return this.txDocumentoses;  
 }  
   
 public void setTxDocumentoses(Set<TxDocumentos> txDocumentoses) {  
 this.txDocumentoses = txDocumentoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\IFTxCincominutalable.java

package mx.mbi.dgc.jpa.entity;  
  
public interface IFTxCincominutalable {  
 TxCincominutal getTxCincominutal();  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\SatImpuestos.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
  
/\*\*  
 \* SatImpuestos generated by hbm2java  
 \*/  
@Entity  
@Table(name="SAT\_IMPUESTOS"  
)  
public class SatImpuestos implements java.io.Serializable {  
  
  
 private BigDecimal impuestoId;  
 private String nombre;  
 private String clave;  
 private Boolean esRetencion;  
 private Boolean esTraslado;  
 private Boolean esFederal;  
 private Boolean activo;  
  
 public SatImpuestos() {  
 }  
  
   
 public SatImpuestos(String nombre) {  
 this.nombre = nombre;  
 }  
 public SatImpuestos(String nombre, String clave, Boolean esRetencion, Boolean esTraslado, Boolean esFederal, Boolean activo) {  
 this.nombre = nombre;  
 this.clave = clave;  
 this.esRetencion = esRetencion;  
 this.esTraslado = esTraslado;  
 this.esFederal = esFederal;  
 this.activo = activo;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="IMPUESTO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getImpuestoId() {  
 return this.impuestoId;  
 }  
   
 public void setImpuestoId(BigDecimal impuestoId) {  
 this.impuestoId = impuestoId;  
 }  
  
   
 @Column(name="NOMBRE", nullable=false, length=60)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="CLAVE", length=10)  
 public String getClave() {  
 return this.clave;  
 }  
   
 public void setClave(String clave) {  
 this.clave = clave;  
 }  
  
   
 @Column(name="ES\_RETENCION", precision=1, scale=0)  
 public Boolean getEsRetencion() {  
 return this.esRetencion;  
 }  
   
 public void setEsRetencion(Boolean esRetencion) {  
 this.esRetencion = esRetencion;  
 }  
  
   
 @Column(name="ES\_TRASLADO", precision=1, scale=0)  
 public Boolean getEsTraslado() {  
 return this.esTraslado;  
 }  
   
 public void setEsTraslado(Boolean esTraslado) {  
 this.esTraslado = esTraslado;  
 }  
  
   
 @Column(name="ES\_FEDERAL", precision=1, scale=0)  
 public Boolean getEsFederal() {  
 return this.esFederal;  
 }  
   
 public void setEsFederal(Boolean esFederal) {  
 this.esFederal = esFederal;  
 }  
  
   
 @Column(name="ACTIVO", precision=1, scale=0)  
 public Boolean getActivo() {  
 return this.activo;  
 }  
   
 public void setActivo(Boolean activo) {  
 this.activo = activo;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\SatMonedas.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
  
/\*\*  
 \* SatMonedas generated by hbm2java  
 \*/  
@Entity  
@Table(name="SAT\_MONEDAS"  
)  
public class SatMonedas implements java.io.Serializable {  
  
  
 private BigDecimal monedaSatId;  
 private String nombre;  
 private String clave;  
 private Byte decimales;  
 private Short porcentajeVariacion;  
 private Boolean activo;  
  
 public SatMonedas() {  
 }  
  
   
 public SatMonedas(String nombre) {  
 this.nombre = nombre;  
 }  
 public SatMonedas(String nombre, String clave, Byte decimales, Short porcentajeVariacion, Boolean activo) {  
 this.nombre = nombre;  
 this.clave = clave;  
 this.decimales = decimales;  
 this.porcentajeVariacion = porcentajeVariacion;  
 this.activo = activo;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="MONEDA\_SAT\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getMonedaSatId() {  
 return this.monedaSatId;  
 }  
   
 public void setMonedaSatId(BigDecimal monedaSatId) {  
 this.monedaSatId = monedaSatId;  
 }  
  
   
 @Column(name="NOMBRE", nullable=false, length=100)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="CLAVE", length=45)  
 public String getClave() {  
 return this.clave;  
 }  
   
 public void setClave(String clave) {  
 this.clave = clave;  
 }  
  
   
 @Column(name="DECIMALES", precision=2, scale=0)  
 public Byte getDecimales() {  
 return this.decimales;  
 }  
   
 public void setDecimales(Byte decimales) {  
 this.decimales = decimales;  
 }  
  
   
 @Column(name="PORCENTAJE\_VARIACION", precision=3, scale=0)  
 public Short getPorcentajeVariacion() {  
 return this.porcentajeVariacion;  
 }  
   
 public void setPorcentajeVariacion(Short porcentajeVariacion) {  
 this.porcentajeVariacion = porcentajeVariacion;  
 }  
  
   
 @Column(name="ACTIVO", precision=1, scale=0)  
 public Boolean getActivo() {  
 return this.activo;  
 }  
   
 public void setActivo(Boolean activo) {  
 this.activo = activo;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\SatPaises.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
  
/\*\*  
 \* SatPaises generated by hbm2java  
 \*/  
@Entity  
@Table(name="SAT\_PAISES"  
)  
public class SatPaises implements java.io.Serializable {  
  
  
 private BigDecimal paisSatId;  
 private String nombre;  
 private String clave;  
 private String formatoCp;  
 private String formatoRit;  
 private String validacionRit;  
 private String agrupaciones;  
 private Boolean activo;  
  
 public SatPaises() {  
 }  
  
   
 public SatPaises(String nombre, String clave) {  
 this.nombre = nombre;  
 this.clave = clave;  
 }  
 public SatPaises(String nombre, String clave, String formatoCp, String formatoRit, String validacionRit, String agrupaciones, Boolean activo) {  
 this.nombre = nombre;  
 this.clave = clave;  
 this.formatoCp = formatoCp;  
 this.formatoRit = formatoRit;  
 this.validacionRit = validacionRit;  
 this.agrupaciones = agrupaciones;  
 this.activo = activo;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="PAIS\_SAT\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getPaisSatId() {  
 return this.paisSatId;  
 }  
   
 public void setPaisSatId(BigDecimal paisSatId) {  
 this.paisSatId = paisSatId;  
 }  
  
   
 @Column(name="NOMBRE", nullable=false, length=60)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="CLAVE", nullable=false, length=60)  
 public String getClave() {  
 return this.clave;  
 }  
   
 public void setClave(String clave) {  
 this.clave = clave;  
 }  
  
   
 @Column(name="FORMATO\_CP", length=60)  
 public String getFormatoCp() {  
 return this.formatoCp;  
 }  
   
 public void setFormatoCp(String formatoCp) {  
 this.formatoCp = formatoCp;  
 }  
  
   
 @Column(name="FORMATO\_RIT", length=100)  
 public String getFormatoRit() {  
 return this.formatoRit;  
 }  
   
 public void setFormatoRit(String formatoRit) {  
 this.formatoRit = formatoRit;  
 }  
  
   
 @Column(name="VALIDACION\_RIT", length=100)  
 public String getValidacionRit() {  
 return this.validacionRit;  
 }  
   
 public void setValidacionRit(String validacionRit) {  
 this.validacionRit = validacionRit;  
 }  
  
   
 @Column(name="AGRUPACIONES", length=60)  
 public String getAgrupaciones() {  
 return this.agrupaciones;  
 }  
   
 public void setAgrupaciones(String agrupaciones) {  
 this.agrupaciones = agrupaciones;  
 }  
  
   
 @Column(name="ACTIVO", precision=1, scale=0)  
 public Boolean getActivo() {  
 return this.activo;  
 }  
   
 public void setActivo(Boolean activo) {  
 this.activo = activo;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\SatProductosClaves.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.Date;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* SatProductosClaves generated by hbm2java  
 \*/  
@Entity  
@Table(name="SAT\_PRODUCTOS\_CLAVES"  
)  
public class SatProductosClaves implements java.io.Serializable {  
  
  
 private BigDecimal satProductosClavesId;  
 private String nombre;  
 private Date fechaIniciaVigencia;  
 private Date fechaFinVigencia;  
 private Boolean incluirIvaTrasladado;  
 private Boolean incluirIepsTrasladado;  
 private String clave;  
 private Boolean activo;  
 private String palabrasSimilares;  
 private Set<CatServicios> catServicioses = new HashSet<CatServicios>(0);  
  
 public SatProductosClaves() {  
 }  
  
   
 public SatProductosClaves(String nombre) {  
 this.nombre = nombre;  
 }  
 public SatProductosClaves(String nombre, Date fechaIniciaVigencia, Date fechaFinVigencia, Boolean incluirIvaTrasladado, Boolean incluirIepsTrasladado, String clave, Boolean activo, String palabrasSimilares, Set<CatServicios> catServicioses) {  
 this.nombre = nombre;  
 this.fechaIniciaVigencia = fechaIniciaVigencia;  
 this.fechaFinVigencia = fechaFinVigencia;  
 this.incluirIvaTrasladado = incluirIvaTrasladado;  
 this.incluirIepsTrasladado = incluirIepsTrasladado;  
 this.clave = clave;  
 this.activo = activo;  
 this.palabrasSimilares = palabrasSimilares;  
 this.catServicioses = catServicioses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="SAT\_PRODUCTOS\_CLAVES\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getSatProductosClavesId() {  
 return this.satProductosClavesId;  
 }  
   
 public void setSatProductosClavesId(BigDecimal satProductosClavesId) {  
 this.satProductosClavesId = satProductosClavesId;  
 }  
  
   
 @Column(name="NOMBRE", nullable=false, length=300)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="FECHA\_INICIA\_VIGENCIA", length=7)  
 public Date getFechaIniciaVigencia() {  
 return this.fechaIniciaVigencia;  
 }  
   
 public void setFechaIniciaVigencia(Date fechaIniciaVigencia) {  
 this.fechaIniciaVigencia = fechaIniciaVigencia;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="FECHA\_FIN\_VIGENCIA", length=7)  
 public Date getFechaFinVigencia() {  
 return this.fechaFinVigencia;  
 }  
   
 public void setFechaFinVigencia(Date fechaFinVigencia) {  
 this.fechaFinVigencia = fechaFinVigencia;  
 }  
  
   
 @Column(name="INCLUIR\_IVA\_TRASLADADO", precision=1, scale=0)  
 public Boolean getIncluirIvaTrasladado() {  
 return this.incluirIvaTrasladado;  
 }  
   
 public void setIncluirIvaTrasladado(Boolean incluirIvaTrasladado) {  
 this.incluirIvaTrasladado = incluirIvaTrasladado;  
 }  
  
   
 @Column(name="INCLUIR\_IEPS\_TRASLADADO", precision=1, scale=0)  
 public Boolean getIncluirIepsTrasladado() {  
 return this.incluirIepsTrasladado;  
 }  
   
 public void setIncluirIepsTrasladado(Boolean incluirIepsTrasladado) {  
 this.incluirIepsTrasladado = incluirIepsTrasladado;  
 }  
  
   
 @Column(name="CLAVE", length=10)  
 public String getClave() {  
 return this.clave;  
 }  
   
 public void setClave(String clave) {  
 this.clave = clave;  
 }  
  
   
 @Column(name="ACTIVO", precision=1, scale=0)  
 public Boolean getActivo() {  
 return this.activo;  
 }  
   
 public void setActivo(Boolean activo) {  
 this.activo = activo;  
 }  
  
   
 @Column(name="PALABRAS\_SIMILARES", length=2000)  
 public String getPalabrasSimilares() {  
 return this.palabrasSimilares;  
 }  
   
 public void setPalabrasSimilares(String palabrasSimilares) {  
 this.palabrasSimilares = palabrasSimilares;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="satProductosClaves")  
 public Set<CatServicios> getCatServicioses() {  
 return this.catServicioses;  
 }  
   
 public void setCatServicioses(Set<CatServicios> catServicioses) {  
 this.catServicioses = catServicioses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\SatProductosUnidades.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.Date;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* SatProductosUnidades generated by hbm2java  
 \*/  
@Entity  
@Table(name="SAT\_PRODUCTOS\_UNIDADES"  
)  
public class SatProductosUnidades implements java.io.Serializable {  
  
  
 private BigDecimal satProductosUnidadesId;  
 private String nombre;  
 private String descripcion;  
 private Date fechaIniciaVigencia;  
 private Date fechaFinVigencia;  
 private Boolean incluirIvaTrasladado;  
 private Boolean incluirIepsTrasladado;  
 private String clave;  
 private String simbolo;  
 private Boolean activo;  
 private String nota;  
 private Set<CatServicios> catServicioses = new HashSet<CatServicios>(0);  
  
 public SatProductosUnidades() {  
 }  
  
   
 public SatProductosUnidades(String nombre, String clave) {  
 this.nombre = nombre;  
 this.clave = clave;  
 }  
 public SatProductosUnidades(String nombre, String descripcion, Date fechaIniciaVigencia, Date fechaFinVigencia, Boolean incluirIvaTrasladado, Boolean incluirIepsTrasladado, String clave, String simbolo, Boolean activo, String nota, Set<CatServicios> catServicioses) {  
 this.nombre = nombre;  
 this.descripcion = descripcion;  
 this.fechaIniciaVigencia = fechaIniciaVigencia;  
 this.fechaFinVigencia = fechaFinVigencia;  
 this.incluirIvaTrasladado = incluirIvaTrasladado;  
 this.incluirIepsTrasladado = incluirIepsTrasladado;  
 this.clave = clave;  
 this.simbolo = simbolo;  
 this.activo = activo;  
 this.nota = nota;  
 this.catServicioses = catServicioses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="SAT\_PRODUCTOS\_UNIDADES\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getSatProductosUnidadesId() {  
 return this.satProductosUnidadesId;  
 }  
   
 public void setSatProductosUnidadesId(BigDecimal satProductosUnidadesId) {  
 this.satProductosUnidadesId = satProductosUnidadesId;  
 }  
  
   
 @Column(name="NOMBRE", nullable=false, length=300)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="DESCRIPCION", length=1000)  
 public String getDescripcion() {  
 return this.descripcion;  
 }  
   
 public void setDescripcion(String descripcion) {  
 this.descripcion = descripcion;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="FECHA\_INICIA\_VIGENCIA", length=7)  
 public Date getFechaIniciaVigencia() {  
 return this.fechaIniciaVigencia;  
 }  
   
 public void setFechaIniciaVigencia(Date fechaIniciaVigencia) {  
 this.fechaIniciaVigencia = fechaIniciaVigencia;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="FECHA\_FIN\_VIGENCIA", length=7)  
 public Date getFechaFinVigencia() {  
 return this.fechaFinVigencia;  
 }  
   
 public void setFechaFinVigencia(Date fechaFinVigencia) {  
 this.fechaFinVigencia = fechaFinVigencia;  
 }  
  
   
 @Column(name="INCLUIR\_IVA\_TRASLADADO", precision=1, scale=0)  
 public Boolean getIncluirIvaTrasladado() {  
 return this.incluirIvaTrasladado;  
 }  
   
 public void setIncluirIvaTrasladado(Boolean incluirIvaTrasladado) {  
 this.incluirIvaTrasladado = incluirIvaTrasladado;  
 }  
  
   
 @Column(name="INCLUIR\_IEPS\_TRASLADADO", precision=1, scale=0)  
 public Boolean getIncluirIepsTrasladado() {  
 return this.incluirIepsTrasladado;  
 }  
   
 public void setIncluirIepsTrasladado(Boolean incluirIepsTrasladado) {  
 this.incluirIepsTrasladado = incluirIepsTrasladado;  
 }  
  
   
 @Column(name="CLAVE", nullable=false, length=10)  
 public String getClave() {  
 return this.clave;  
 }  
   
 public void setClave(String clave) {  
 this.clave = clave;  
 }  
  
   
 @Column(name="SIMBOLO", length=50)  
 public String getSimbolo() {  
 return this.simbolo;  
 }  
   
 public void setSimbolo(String simbolo) {  
 this.simbolo = simbolo;  
 }  
  
   
 @Column(name="ACTIVO", precision=1, scale=0)  
 public Boolean getActivo() {  
 return this.activo;  
 }  
   
 public void setActivo(Boolean activo) {  
 this.activo = activo;  
 }  
  
   
 @Column(name="NOTA", length=2000)  
 public String getNota() {  
 return this.nota;  
 }  
   
 public void setNota(String nota) {  
 this.nota = nota;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="satProductosUnidades")  
 public Set<CatServicios> getCatServicioses() {  
 return this.catServicioses;  
 }  
   
 public void setCatServicioses(Set<CatServicios> catServicioses) {  
 this.catServicioses = catServicioses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\SatTasaCuota.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.Date;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* SatTasaCuota generated by hbm2java  
 \*/  
@Entity  
@Table(name="SAT\_TASA\_CUOTA"  
)  
public class SatTasaCuota implements java.io.Serializable {  
  
  
 private BigDecimal tasaCuotaId;  
 private String rango;  
 private BigDecimal valorMinimo;  
 private BigDecimal valorMaximo;  
 private BigDecimal impuestoId;  
 private String impuesto;  
 private String factor;  
 private String traslado;  
 private String retencion;  
 private Date fechaInicioVigencia;  
 private Date fechaFinVigencia;  
  
 public SatTasaCuota() {  
 }  
  
   
 public SatTasaCuota(String rango, BigDecimal valorMaximo, String impuesto, String factor) {  
 this.rango = rango;  
 this.valorMaximo = valorMaximo;  
 this.impuesto = impuesto;  
 this.factor = factor;  
 }  
 public SatTasaCuota(String rango, BigDecimal valorMinimo, BigDecimal valorMaximo, BigDecimal impuestoId, String impuesto, String factor, String traslado, String retencion, Date fechaInicioVigencia, Date fechaFinVigencia) {  
 this.rango = rango;  
 this.valorMinimo = valorMinimo;  
 this.valorMaximo = valorMaximo;  
 this.impuestoId = impuestoId;  
 this.impuesto = impuesto;  
 this.factor = factor;  
 this.traslado = traslado;  
 this.retencion = retencion;  
 this.fechaInicioVigencia = fechaInicioVigencia;  
 this.fechaFinVigencia = fechaFinVigencia;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="TASA\_CUOTA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getTasaCuotaId() {  
 return this.tasaCuotaId;  
 }  
   
 public void setTasaCuotaId(BigDecimal tasaCuotaId) {  
 this.tasaCuotaId = tasaCuotaId;  
 }  
  
   
 @Column(name="RANGO", nullable=false, length=30)  
 public String getRango() {  
 return this.rango;  
 }  
   
 public void setRango(String rango) {  
 this.rango = rango;  
 }  
  
   
 @Column(name="VALOR\_MINIMO", precision=20, scale=6)  
 public BigDecimal getValorMinimo() {  
 return this.valorMinimo;  
 }  
   
 public void setValorMinimo(BigDecimal valorMinimo) {  
 this.valorMinimo = valorMinimo;  
 }  
  
   
 @Column(name="VALOR\_MAXIMO", nullable=false, precision=20, scale=6)  
 public BigDecimal getValorMaximo() {  
 return this.valorMaximo;  
 }  
   
 public void setValorMaximo(BigDecimal valorMaximo) {  
 this.valorMaximo = valorMaximo;  
 }  
  
   
 @Column(name="IMPUESTO\_ID", precision=20, scale=0)  
 public BigDecimal getImpuestoId() {  
 return this.impuestoId;  
 }  
   
 public void setImpuestoId(BigDecimal impuestoId) {  
 this.impuestoId = impuestoId;  
 }  
  
   
 @Column(name="IMPUESTO", nullable=false, length=30)  
 public String getImpuesto() {  
 return this.impuesto;  
 }  
   
 public void setImpuesto(String impuesto) {  
 this.impuesto = impuesto;  
 }  
  
   
 @Column(name="FACTOR", nullable=false, length=30)  
 public String getFactor() {  
 return this.factor;  
 }  
   
 public void setFactor(String factor) {  
 this.factor = factor;  
 }  
  
   
 @Column(name="TRASLADO", length=20)  
 public String getTraslado() {  
 return this.traslado;  
 }  
   
 public void setTraslado(String traslado) {  
 this.traslado = traslado;  
 }  
  
   
 @Column(name="RETENCION", length=20)  
 public String getRetencion() {  
 return this.retencion;  
 }  
   
 public void setRetencion(String retencion) {  
 this.retencion = retencion;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="FECHA\_INICIO\_VIGENCIA", length=7)  
 public Date getFechaInicioVigencia() {  
 return this.fechaInicioVigencia;  
 }  
   
 public void setFechaInicioVigencia(Date fechaInicioVigencia) {  
 this.fechaInicioVigencia = fechaInicioVigencia;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="FECHA\_FIN\_VIGENCIA", length=7)  
 public Date getFechaFinVigencia() {  
 return this.fechaFinVigencia;  
 }  
   
 public void setFechaFinVigencia(Date fechaFinVigencia) {  
 this.fechaFinVigencia = fechaFinVigencia;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\SatTipoFactor.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
  
/\*\*  
 \* SatTipoFactor generated by hbm2java  
 \*/  
@Entity  
@Table(name="SAT\_TIPO\_FACTOR"  
)  
public class SatTipoFactor implements java.io.Serializable {  
  
  
 private BigDecimal tipoFactorId;  
 private String nombre;  
 private boolean activo;  
  
 public SatTipoFactor() {  
 }  
  
 public SatTipoFactor(String nombre, boolean activo) {  
 this.nombre = nombre;  
 this.activo = activo;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="TIPO\_FACTOR\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getTipoFactorId() {  
 return this.tipoFactorId;  
 }  
   
 public void setTipoFactorId(BigDecimal tipoFactorId) {  
 this.tipoFactorId = tipoFactorId;  
 }  
  
   
 @Column(name="NOMBRE", nullable=false, length=60)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="ACTIVO", nullable=false, precision=1, scale=0)  
 public boolean isActivo() {  
 return this.activo;  
 }  
   
 public void setActivo(boolean activo) {  
 this.activo = activo;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\SatTipoRelacion.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.util.Date;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* SatTipoRelacion generated by hbm2java  
 \*/  
@Entity  
@Table(name="SAT\_TIPO\_RELACION"  
)  
public class SatTipoRelacion implements java.io.Serializable {  
  
  
 private String nombre;  
 private Boolean activo;  
 private Date fechaInicioVigencia;  
 private Date fechaFinVigencia;  
 private String tipoRelacion;  
  
 public SatTipoRelacion() {  
 }  
  
 public SatTipoRelacion(Boolean activo, Date fechaInicioVigencia, Date fechaFinVigencia, String tipoRelacion) {  
 this.activo = activo;  
 this.fechaInicioVigencia = fechaInicioVigencia;  
 this.fechaFinVigencia = fechaFinVigencia;  
 this.tipoRelacion = tipoRelacion;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="NOMBRE", unique=true, nullable=false, length=100)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
   
 @Column(name="ACTIVO", precision=1, scale=0)  
 public Boolean getActivo() {  
 return this.activo;  
 }  
   
 public void setActivo(Boolean activo) {  
 this.activo = activo;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="FECHA\_INICIO\_VIGENCIA", length=7)  
 public Date getFechaInicioVigencia() {  
 return this.fechaInicioVigencia;  
 }  
   
 public void setFechaInicioVigencia(Date fechaInicioVigencia) {  
 this.fechaInicioVigencia = fechaInicioVigencia;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="FECHA\_FIN\_VIGENCIA", length=7)  
 public Date getFechaFinVigencia() {  
 return this.fechaFinVigencia;  
 }  
   
 public void setFechaFinVigencia(Date fechaFinVigencia) {  
 this.fechaFinVigencia = fechaFinVigencia;  
 }  
  
   
 @Column(name="TIPO\_RELACION", length=20)  
 public String getTipoRelacion() {  
 return this.tipoRelacion;  
 }  
   
 public void setTipoRelacion(String tipoRelacion) {  
 this.tipoRelacion = tipoRelacion;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\SatUsoCfdi.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.util.Date;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* SatUsoCfdi generated by hbm2java  
 \*/  
@Entity  
@Table(name="SAT\_USO\_CFDI"  
)  
public class SatUsoCfdi implements java.io.Serializable {  
  
  
 private String descripcion;  
 private boolean fisica;  
 private boolean moral;  
 private Date fechaInicioVigencia;  
 private Date fechaFinVigencia;  
 private String usoCfdiCve;  
  
 public SatUsoCfdi() {  
 }  
  
   
 public SatUsoCfdi(boolean fisica, boolean moral) {  
 this.fisica = fisica;  
 this.moral = moral;  
 }  
 public SatUsoCfdi(boolean fisica, boolean moral, Date fechaInicioVigencia, Date fechaFinVigencia, String usoCfdiCve) {  
 this.fisica = fisica;  
 this.moral = moral;  
 this.fechaInicioVigencia = fechaInicioVigencia;  
 this.fechaFinVigencia = fechaFinVigencia;  
 this.usoCfdiCve = usoCfdiCve;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="DESCRIPCION", unique=true, nullable=false, length=200)  
 public String getDescripcion() {  
 return this.descripcion;  
 }  
   
 public void setDescripcion(String descripcion) {  
 this.descripcion = descripcion;  
 }  
  
   
 @Column(name="FISICA", nullable=false, precision=1, scale=0)  
 public boolean isFisica() {  
 return this.fisica;  
 }  
   
 public void setFisica(boolean fisica) {  
 this.fisica = fisica;  
 }  
  
   
 @Column(name="MORAL", nullable=false, precision=1, scale=0)  
 public boolean isMoral() {  
 return this.moral;  
 }  
   
 public void setMoral(boolean moral) {  
 this.moral = moral;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="FECHA\_INICIO\_VIGENCIA", length=7)  
 public Date getFechaInicioVigencia() {  
 return this.fechaInicioVigencia;  
 }  
   
 public void setFechaInicioVigencia(Date fechaInicioVigencia) {  
 this.fechaInicioVigencia = fechaInicioVigencia;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="FECHA\_FIN\_VIGENCIA", length=7)  
 public Date getFechaFinVigencia() {  
 return this.fechaFinVigencia;  
 }  
   
 public void setFechaFinVigencia(Date fechaFinVigencia) {  
 this.fechaFinVigencia = fechaFinVigencia;  
 }  
  
   
 @Column(name="USO\_CFDI\_CVE", length=20)  
 public String getUsoCfdiCve() {  
 return this.usoCfdiCve;  
 }  
   
 public void setUsoCfdiCve(String usoCfdiCve) {  
 this.usoCfdiCve = usoCfdiCve;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxAdjuntos.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
  
import java.math.BigDecimal;  
  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.Table;  
  
import javax.validation.constraints.Size;  
  
import javax.xml.bind.annotation.XmlRootElement;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "TX\_ADJUNTOS")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "TxAdjuntos.findAll", query = "SELECT t FROM TxAdjuntos t")  
 , @NamedQuery(name = "TxAdjuntos.findByAdjuntoId", query = "SELECT t FROM TxAdjuntos t WHERE t.adjuntoId = :adjuntoId")  
 , @NamedQuery(name = "TxAdjuntos.findByReferencia", query = "SELECT t FROM TxAdjuntos t WHERE t.referencia = :referencia")})  
public class TxAdjuntos implements Serializable {  
  
 @JoinColumn(name = "EVENTO\_ID", referencedColumnName = "EVENTO\_ID")  
 @ManyToOne  
 private TxEvento eventoId;  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id @GeneratedValue(strategy=IDENTITY)  
 @Basic(optional = false)  
 @Column(name = "ADJUNTO\_ID")  
 private BigDecimal adjuntoId;  
 @Basic(optional = false)  
 @Size(min = 1, max = 250)  
 @Column(name = "REFERENCIA")  
 private String referencia;  
 @JoinColumn(name = "TIPO\_DOCUMENTO\_ID", referencedColumnName = "TIPO\_DOCUMENTO\_ID")  
 @ManyToOne(optional = false)  
 private CatTipoDocumento tipoDocumentoId;  
   
  
 public TxAdjuntos() {  
 }  
  
 public TxAdjuntos(BigDecimal adjuntoId) {  
 this.adjuntoId = adjuntoId;  
 }  
  
 public TxAdjuntos(BigDecimal adjuntoId, String referencia) {  
 this.adjuntoId = adjuntoId;  
 this.referencia = referencia;  
 }  
  
 public BigDecimal getAdjuntoId() {  
 return adjuntoId;  
 }  
  
 public void setAdjuntoId(BigDecimal adjuntoId) {  
 this.adjuntoId = adjuntoId;  
 }  
  
 public String getReferencia() {  
 return referencia;  
 }  
  
 public void setReferencia(String referencia) {  
 this.referencia = referencia;  
 }  
  
 public CatTipoDocumento getTipoDocumentoId() {  
 return tipoDocumentoId;  
 }  
  
 public void setTipoDocumentoId(CatTipoDocumento tipoDocumentoId) {  
 this.tipoDocumentoId = tipoDocumentoId;  
 }  
  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (adjuntoId != null ? adjuntoId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof TxAdjuntos)) {  
 return false;  
 }  
 TxAdjuntos other = (TxAdjuntos) object;  
 if ((this.adjuntoId == null && other.adjuntoId != null) || (this.adjuntoId != null && !this.adjuntoId.equals(other.adjuntoId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.TxAdjuntos[ adjuntoId=" + adjuntoId + " ]";  
 }  
  
 public TxEvento getEventoId() {  
 return eventoId;  
 }  
  
 public void setEventoId(TxEvento eventoId) {  
 this.eventoId = eventoId;  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxBitacora.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
  
import java.math.BigDecimal;  
  
import java.util.Date;  
import java.util.Set;  
  
import javax.persistence.Basic;  
import javax.persistence.CascadeType;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
import javax.validation.constraints.NotNull;  
  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlTransient;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "TX\_BITACORA")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "TxBitacora.findAll", query = "SELECT t FROM TxBitacora t")  
 , @NamedQuery(name = "TxBitacora.findByBitacoraId", query = "SELECT t FROM TxBitacora t WHERE t.bitacoraId = :bitacoraId")  
 , @NamedQuery(name = "TxBitacora.findByFecha", query = "SELECT t FROM TxBitacora t WHERE t.fecha = :fecha")})  
public class TxBitacora implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "BITACORA\_ID")  
 private BigDecimal bitacoraId;  
 @Basic(optional = false)  
 @NotNull  
 @Column(name = "FECHA")  
 @Temporal(TemporalType.DATE)  
 private Date fecha;  
 @JoinColumn(name = "ESTATUS\_BITACORA\_ID", referencedColumnName = "ESTATUS\_BITACORA\_ID")  
 @ManyToOne(optional = false)  
 private CatEstatusBitacora estatusBitacoraId;  
 @JoinColumn(name = "PLANTA\_ID", referencedColumnName = "PLANTA\_ID")  
 @ManyToOne(optional = false)  
 private CatPlanta plantaId;  
 @OneToMany(cascade = CascadeType.ALL, mappedBy = "bitacoraId")  
 private Set<TxEvento> txEventoSet;  
  
 public TxBitacora() {  
 }  
  
 public TxBitacora(BigDecimal bitacoraId) {  
 this.bitacoraId = bitacoraId;  
 }  
  
 public TxBitacora(BigDecimal bitacoraId, Date fecha) {  
 this.bitacoraId = bitacoraId;  
 this.fecha = fecha;  
 }  
  
 public BigDecimal getBitacoraId() {  
 return bitacoraId;  
 }  
  
 public void setBitacoraId(BigDecimal bitacoraId) {  
 this.bitacoraId = bitacoraId;  
 }  
  
 public Date getFecha() {  
 return fecha;  
 }  
  
 public void setFecha(Date fecha) {  
 this.fecha = fecha;  
 }  
  
 public CatEstatusBitacora getEstatusBitacoraId() {  
 return estatusBitacoraId;  
 }  
  
 public void setEstatusBitacoraId(CatEstatusBitacora estatusBitacoraId) {  
 this.estatusBitacoraId = estatusBitacoraId;  
 }  
  
 public CatPlanta getPlantaId() {  
 return plantaId;  
 }  
  
 public void setPlantaId(CatPlanta plantaId) {  
 this.plantaId = plantaId;  
 }  
  
  
 @XmlTransient  
 public Set<TxEvento> getTxEventoSet() {  
 return txEventoSet;  
 }  
  
 public void setTxEventoSet(Set<TxEvento> txEventoSet) {  
 this.txEventoSet = txEventoSet;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (bitacoraId != null ? bitacoraId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof TxBitacora)) {  
 return false;  
 }  
 TxBitacora other = (TxBitacora) object;  
 if ((this.bitacoraId == null && other.bitacoraId != null) || (this.bitacoraId != null && !this.bitacoraId.equals(other.bitacoraId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.TxBitacora[ bitacoraId=" + bitacoraId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxCincominutal.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import java.util.HashSet;  
import java.util.Set;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.persistence.UniqueConstraint;  
  
/\*\*  
 \* TxCincominutal generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_CINCOMINUTAL"  
 , uniqueConstraints = @UniqueConstraint(columnNames="MES\_DIA\_FIRMA\_ID")   
)  
public class TxCincominutal implements java.io.Serializable {  
 private BigDecimal cincominutalId;  
 private CatPlanta catPlanta;  
 private CatMesDiaFirma catMesDiaFirma;  
 private BigDecimal cargaCincominutal;  
 private BigDecimal energiaActiva;  
 private BigDecimal energiaReactiva;  
 private BigDecimal factorDePotencia;  
 private CatCombustible catCombustible;  
 private Set<TxPerfilesVariable> txPerfilesVariables = new HashSet<TxPerfilesVariable>(0);  
 private Set<TxCincominutalVar> txCincominutalVars = new HashSet<TxCincominutalVar>(0);  
 private Set<TxProduccion> txProduccions = new HashSet<TxProduccion>(0);  
 private CatEstatusVariable catEstatusVariable;  
  
 public TxCincominutal() {  
 }  
  
   
 public TxCincominutal(CatPlanta catPlanta, CatMesDiaFirma catMesDiaFirma, BigDecimal cargaCincominutal, BigDecimal energiaActiva, BigDecimal energiaReactiva) {  
 this.catPlanta = catPlanta;  
 this.catMesDiaFirma = catMesDiaFirma;  
 this.cargaCincominutal = cargaCincominutal;  
 this.energiaActiva = energiaActiva;  
 this.energiaReactiva = energiaReactiva;  
 }  
 public TxCincominutal(CatPlanta catPlanta, CatMesDiaFirma catMesDiaFirma, BigDecimal cargaCincominutal, BigDecimal energiaActiva, BigDecimal energiaReactiva, BigDecimal factorDePotencia, CatCombustible catCombustible, Set<TxPerfilesVariable> txPerfilesVariables, Set<TxCincominutalVar> txCincominutalVars, Set<TxProduccion> txProduccions) {  
 this.catPlanta = catPlanta;  
 this.catMesDiaFirma = catMesDiaFirma;  
 this.cargaCincominutal = cargaCincominutal;  
 this.energiaActiva = energiaActiva;  
 this.energiaReactiva = energiaReactiva;  
 this.factorDePotencia = factorDePotencia;  
 this.catCombustible = catCombustible;  
 this.txPerfilesVariables = txPerfilesVariables;  
 this.txCincominutalVars = txCincominutalVars;  
 this.txProduccions = txProduccions;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="CINCOMINUTAL\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getCincominutalId() {  
 return this.cincominutalId;  
 }  
   
 public void setCincominutalId(BigDecimal cincominutalId) {  
 this.cincominutalId = cincominutalId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="PLANTA\_ID", nullable=false)  
 public CatPlanta getCatPlanta() {  
 return this.catPlanta;  
 }  
   
 public void setCatPlanta(CatPlanta catPlanta) {  
 this.catPlanta = catPlanta;  
 }  
  
 @ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="MES\_DIA\_FIRMA\_ID", unique=true, nullable=false)  
 public CatMesDiaFirma getCatMesDiaFirma() {  
 return this.catMesDiaFirma;  
 }  
   
 public void setCatMesDiaFirma(CatMesDiaFirma catMesDiaFirma) {  
 this.catMesDiaFirma = catMesDiaFirma;  
 }  
  
   
 @Column(name="CARGA\_CINCOMINUTAL", nullable=false, precision=22, scale=0)  
 public BigDecimal getCargaCincominutal() {  
 return this.cargaCincominutal;  
 }  
   
 public void setCargaCincominutal(BigDecimal cargaCincominutal) {  
 this.cargaCincominutal = cargaCincominutal;  
 }  
  
   
 @Column(name="ENERGIA\_ACTIVA", nullable=false, precision=22, scale=0)  
 public BigDecimal getEnergiaActiva() {  
 return this.energiaActiva;  
 }  
   
 public void setEnergiaActiva(BigDecimal energiaActiva) {  
 this.energiaActiva = energiaActiva;  
 }  
  
   
 @Column(name="ENERGIA\_REACTIVA", nullable=false, precision=22, scale=0)  
 public BigDecimal getEnergiaReactiva() {  
 return this.energiaReactiva;  
 }  
   
 public void setEnergiaReactiva(BigDecimal energiaReactiva) {  
 this.energiaReactiva = energiaReactiva;  
 }  
  
   
 @Column(name="FACTOR\_DE\_POTENCIA", precision=22, scale=0)  
 public BigDecimal getFactorDePotencia() {  
 return this.factorDePotencia;  
 }  
   
 public void setFactorDePotencia(BigDecimal factorDePotencia) {  
 this.factorDePotencia = factorDePotencia;  
 }  
  
   
 @ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="COMBUSTIBLE\_ID", nullable=false)  
 public CatCombustible getCatCombustible() {  
 return this.catCombustible;  
 }  
   
 public void setCatCombustible(CatCombustible catCombustible) {  
 this.catCombustible = catCombustible;  
 }  
  
 @OneToMany(fetch=FetchType.LAZY, mappedBy="txCincominutal")  
 public Set<TxPerfilesVariable> getTxPerfilesVariables() {  
 return this.txPerfilesVariables;  
 }  
   
 public void setTxPerfilesVariables(Set<TxPerfilesVariable> txPerfilesVariables) {  
 this.txPerfilesVariables = txPerfilesVariables;  
 }  
  
   
 @OneToMany(fetch=FetchType.LAZY, mappedBy="txCincominutal")  
 public Set<TxCincominutalVar> getTxCincominutalVars() {  
 return this.txCincominutalVars;  
 }  
   
 public void setTxCincominutalVars(Set<TxCincominutalVar> txCincominutalVars) {  
 this.txCincominutalVars = txCincominutalVars;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="txCincominutal")  
 public Set<TxProduccion> getTxProduccions() {  
 return this.txProduccions;  
 }  
   
 public void setTxProduccions(Set<TxProduccion> txProduccions) {  
 this.txProduccions = txProduccions;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxCincominutalVar.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import java.util.Date;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* TxCincominutalVar generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_CINCOMINUTAL\_VAR")  
public class TxCincominutalVar implements java.io.Serializable, IFTxCincominutalable, Comparable {  
  
 @Override  
 public int compareTo(Object object) {  
 int retorno = 0;  
 TxCincominutalVar actual = (TxCincominutalVar)object;  
 if(actual.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId().intValue() >= this.getTxCincominutal().getCatMesDiaFirma().getMesDiaFirmaId().intValue()){  
 retorno= 1;  
 }  
 return retorno;  
 }  
  
 private BigDecimal cincominutalVarId;  
 private CatVariable catVariable;  
 private TxCincominutal txCincominutal;  
 private String valorOriginal;  
 private Date fechaCarga;  
 private BigDecimal estatusDf;  
 private BigDecimal estatusCf;  
 private BigDecimal valorFormato;  
 private BigDecimal estatusDn;  
 private Date fechaDn;  
 private BigDecimal estatusCn;  
 private Date fechaCn;  
 private BigDecimal valorNorma;  
 private BigDecimal fcPrimera;  
 private Date fechaPc;  
 private BigDecimal fcSegunda;  
 private Date fechaSc;  
 private Date fechaUm;  
  
 public TxCincominutalVar() {  
 }  
  
   
 public TxCincominutalVar(CatVariable catVariable, TxCincominutal txCincominutal, String valorOriginal, Date fechaCarga) {  
 this.catVariable = catVariable;  
 this.txCincominutal = txCincominutal;  
 this.valorOriginal = valorOriginal;  
 this.fechaCarga = fechaCarga;  
 }  
 public TxCincominutalVar(CatVariable catVariable, TxCincominutal txCincominutal, String valorOriginal, Date fechaCarga, BigDecimal estatusDf, BigDecimal estatusCf, BigDecimal valorFormato, BigDecimal estatusDn, Date fechaDn, BigDecimal estatusCn, Date fechaCn, BigDecimal valorNorma, BigDecimal fcPrimera, Date fechaPc, BigDecimal fcSegunda, Date fechaSc, Date fechaUm) {  
 this.catVariable = catVariable;  
 this.txCincominutal = txCincominutal;  
 this.valorOriginal = valorOriginal;  
 this.fechaCarga = fechaCarga;  
 this.estatusDf = estatusDf;  
 this.estatusCf = estatusCf;  
 this.valorFormato = valorFormato;  
 this.estatusDn = estatusDn;  
 this.fechaDn = fechaDn;  
 this.estatusCn = estatusCn;  
 this.fechaCn = fechaCn;  
 this.valorNorma = valorNorma;  
 this.fcPrimera = fcPrimera;  
 this.fechaPc = fechaPc;  
 this.fcSegunda = fcSegunda;  
 this.fechaSc = fechaSc;  
 this.fechaUm = fechaUm;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="CINCOMINUTAL\_VAR\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getCincominutalVarId() {  
 return this.cincominutalVarId;  
 }  
   
 public void setCincominutalVarId(BigDecimal cincominutalVarId) {  
 this.cincominutalVarId = cincominutalVarId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="VARIABLE\_ID", nullable=false)  
 public CatVariable getCatVariable() {  
 return this.catVariable;  
 }  
   
 public void setCatVariable(CatVariable catVariable) {  
 this.catVariable = catVariable;  
 }  
  
 @ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="CINCOMINUTAL\_ID", nullable=false)  
 public TxCincominutal getTxCincominutal() {  
 return this.txCincominutal;  
 }  
   
 public void setTxCincominutal(TxCincominutal txCincominutal) {  
 this.txCincominutal = txCincominutal;  
 }  
  
   
 @Column(name="VALOR\_ORIGINAL", nullable=false, length=100)  
 public String getValorOriginal() {  
 return this.valorOriginal;  
 }  
   
 public void setValorOriginal(String valorOriginal) {  
 this.valorOriginal = valorOriginal;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="FECHA\_CARGA", nullable=false, length=11)  
 public Date getFechaCarga() {  
 return this.fechaCarga;  
 }  
   
 public void setFechaCarga(Date fechaCarga) {  
 this.fechaCarga = fechaCarga;  
 }  
  
   
 @Column(name="ESTATUS\_DF", precision=22, scale=0)  
 public BigDecimal getEstatusDf() {  
 return this.estatusDf;  
 }  
   
 public void setEstatusDf(BigDecimal estatusDf) {  
 this.estatusDf = estatusDf;  
 }  
  
   
 @Column(name="ESTATUS\_CF", precision=22, scale=0)  
 public BigDecimal getEstatusCf() {  
 return this.estatusCf;  
 }  
   
 public void setEstatusCf(BigDecimal estatusCf) {  
 this.estatusCf = estatusCf;  
 }  
  
   
 @Column(name="VALOR\_FORMATO", precision=22, scale=0)  
 public BigDecimal getValorFormato() {  
 return this.valorFormato;  
 }  
   
 public void setValorFormato(BigDecimal valorFormato) {  
 this.valorFormato = valorFormato;  
 }  
  
   
 @Column(name="ESTATUS\_DN", precision=22, scale=0)  
 public BigDecimal getEstatusDn() {  
 return this.estatusDn;  
 }  
   
 public void setEstatusDn(BigDecimal estatusDn) {  
 this.estatusDn = estatusDn;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="FECHA\_DN", length=11)  
 public Date getFechaDn() {  
 return this.fechaDn;  
 }  
   
 public void setFechaDn(Date fechaDn) {  
 this.fechaDn = fechaDn;  
 }  
  
   
 @Column(name="ESTATUS\_CN", precision=22, scale=0)  
 public BigDecimal getEstatusCn() {  
 return this.estatusCn;  
 }  
   
 public void setEstatusCn(BigDecimal estatusCn) {  
 this.estatusCn = estatusCn;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="FECHA\_CN", length=11)  
 public Date getFechaCn() {  
 return this.fechaCn;  
 }  
   
 public void setFechaCn(Date fechaCn) {  
 this.fechaCn = fechaCn;  
 }  
  
   
 @Column(name="VALOR\_NORMA", precision=22, scale=0)  
 public BigDecimal getValorNorma() {  
 return this.valorNorma;  
 }  
   
 public void setValorNorma(BigDecimal valorNorma) {  
 this.valorNorma = valorNorma;  
 }  
  
   
 @Column(name="FC\_PRIMERA", precision=22, scale=0)  
 public BigDecimal getFcPrimera() {  
 return this.fcPrimera;  
 }  
   
 public void setFcPrimera(BigDecimal fcPrimera) {  
 this.fcPrimera = fcPrimera;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="FECHA\_PC", length=11)  
 public Date getFechaPc() {  
 return this.fechaPc;  
 }  
   
 public void setFechaPc(Date fechaPc) {  
 this.fechaPc = fechaPc;  
 }  
  
   
 @Column(name="FC\_SEGUNDA", precision=22, scale=0)  
 public BigDecimal getFcSegunda() {  
 return this.fcSegunda;  
 }  
   
 public void setFcSegunda(BigDecimal fcSegunda) {  
 this.fcSegunda = fcSegunda;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="FECHA\_SC", length=11)  
 public Date getFechaSc() {  
 return this.fechaSc;  
 }  
   
 public void setFechaSc(Date fechaSc) {  
 this.fechaSc = fechaSc;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="FECHA\_UM", length=11)  
 public Date getFechaUm() {  
 return this.fechaUm;  
 }  
   
 public void setFechaUm(Date fechaUm) {  
 this.fechaUm = fechaUm;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxCrudos.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.Date;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* TxCrudos generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_CRUDOS"  
)  
public class TxCrudos implements java.io.Serializable {  
  
  
 private BigDecimal crudosId;  
 private String dia;  
 private String minuto;  
 private String fuelSelect;  
 private String hourlyLoadInUse;  
 private String line1WattHourMas5min;  
 private String line2WattHourMas5min;  
 private String line1WattHourMenos5min;  
 private String line2WattHourMenos5min;  
 private String line1WattHourMas5minBackup;  
 private String line2WattHourMas5minBackup;  
 private String line1WattHourMenos5minBackup;  
 private String line2WattHourMenos5minBackup;  
 private String gt1ActivePower;  
 private String gt2ActivePower;  
 private String stActivePower;  
 private String gt1ReactivePower;  
 private String gt2ReactivePower;  
 private String stReactivePower;  
 private String line1WattInstant;  
 private String line2WattInstant;  
 private String line1VarInstant;  
 private String line2VarInstant;  
 private String plantNetWattHourCalculationFromGross;  
 private String stgWattHour;  
 private String stgVarHour;  
 private String no1GtgWattHour;  
 private String no1GtgVarHour;  
 private String no1GtgUnitHouseLoad;  
 private String no2GtgWattHour;  
 private String no2GtgVarHour;  
 private String no2GtgUnitHouseLoad;  
 private String ambientTemperatureInUse;  
 private String ambientTemperature1;  
 private String ambientTemperature2;  
 private String relativeHumidityInUse;  
 private String relativeHumidity1;  
 private String relativeHumidity2;  
 private String ambientPressureInUse;  
 private String ambientPressure1;  
 private String ambientPressure2;  
 private String condenserVacuumInUse;  
 private String condenserVacuum1;  
 private String condenserVacuum2;  
 private String condenserVacuumMonitoring;  
 private String fuelGasHigherHeatingValue;  
 private String fuelGasLowerHeatingValue;  
 private String fuelGasHigherHeatingValue2;  
 private String fuelGasLowerHeatingValue2;  
 private String fuelOilHigherHeatingValeu;  
 private String fuelOilLowerHeatingValue;  
 private String fuelOilHigherHeatingValeu2;  
 private String fuelOilLowerHeatingValue2;  
 private String fuelOilSpecificGravity;  
 private String fuelGasFlowCurrentMasFlow;  
 private String strm1CurrentVolumeFlowRate;  
 private String fuelGasFlow5minAccumulatedValue;  
 private String oilFlowGt1Gt2;  
 private String gt1OilFlowCurrentValue;  
 private String gt2OilFlowCurrentValue;  
 private String oilFlowGt1Gt22;  
 private String gt1OilFlow5minAccumulatedValue;  
 private String gt2OilFlow5minAccumulatedValue;  
 private String overallPowerFactor;  
 private String netPowerOf5min5minAccumulatedWithGasFiring;  
 private String netPowerOf5min5minAccumulatedWithOilFiring;  
 private String loadPercentageAtGas;  
 private String loadPercentageAtOil;  
 private String ctungAtYearAverageDesignConditionGas;  
 private String ctungAtYearAverageDesignConditionOil;  
 private String correctedCtungGas;  
 private String correctedCtungOil;  
 private String fuelGasPressure;  
 private String gasTemperature;  
 private String loadDemand;  
 private String eh512RollingAverage;  
 private String ambientTemperature12RollingAverage;  
 private String relativeHumidity12RollingAverage;  
 private String ambientPressure12RollingAverage;  
 private String condenserVacuum12RollingAverage;  
 private String fuelGasFlow12RollingAverage;  
 private String fuelGasFlow12RollingAverage2;  
 private String fuelGasPressure12RollingAverage;  
 private String overallPowerFactor12RollingAverage;  
 private String fuelOilFlow12RollingAverage;  
 private String fuelOilFlow12RollingAverage2;  
 private String correctedCtung12RollingWeightedAverageGas;  
 private String correctedCtung12RollingWeightedAverageOil;  
 private String fuelGasHigherHeatingValue12RollingAverage;  
 private String fuelGasLowerHeatingValue12RollingAverage;  
 private String fuelGasHigherHeatingValue12RollingAverage2;  
 private String fuelGasLowerHeatingValue12RollingAverage2;  
 private String fuelOilHigherHeatingValue12RollingAverage;  
 private String fuelOilLowerHeatingValue12RollingAverage;  
 private String fuelOilHigherHeatingValue12RollingAverage2;  
 private String fuelOilLowerHeatingValue12RollingAverage2;  
 private String numberOfMinutesOfOperatingAtGasFiring;  
 private String numberOfMinutesOfOperatingAtOilFiring;  
 private String eh12RollingAverage;  
 private String egh12RollingAverage;  
 private String edh12RollingAverage;  
 private String nhovgOperatingHourAtNoLoadBurningGas;  
 private String nhovdOperatingHourAtNoLoadBurningOil;  
 private String gt1OperatingHour;  
 private String gt2OperatingHour;  
 private String netOutputCorrectedToSummerDesignCfe;  
 private String kcactualGas;  
 private String kcactualOil;  
 private String spare;  
 private String gt152gOn;  
 private String gt252gOn;  
 private String EA1010400kvSSCbGt1Abn;  
 private String EA1010400kvSSCbGt1Close;  
 private String EA1010400kvSSCbGt1Open;  
 private String EA1020400kvSSCbGt2Abn;  
 private String EA1020400kvSSCbGt2Close;  
 private String EA1020400kvSSCbGt2Open;  
 private String EA1030400kvSSCbStClose;  
 private String EA1030400kvSSCbStOpen;  
 private String gt1Synchronize;  
 private String gt2Synchronize;  
 private String allTariffMeterAbnormal;  
 private String gt1TransmitterWatAndVarAbnormal;  
 private String gt2TransmitterWatAndVarAbnormal;  
 private String stTransmitterWatAndVarAbnormal;  
 private String no1TempTransmitterAbnormal;  
 private String no2TempTransmitterAbnormal;  
 private String tempTransmitterSelect0No21No1;  
 private String no1HumidityTransmitterAbnormal;  
 private String no2HumidityTransmitterAbnormal;  
 private String humidityTransmitterSelect0No21No1;  
 private String no1AmbientPressureTransmitterAbnormal;  
 private String no2AmbientPressureTransmitterAbnormal;  
 private String ambientPressureTransmitterSelect0No21No1;  
 private String no1CondVacuumTransmitterAbnormal;  
 private String no2CondVacuumTransmitterAbnormal;  
 private String condVacuumTransmitterSelect0No21No1;  
 private String gt1GasSelect;  
 private String gt2GasSelect;  
 private String gt1OilSelect;  
 private String gt2OilSelect;  
 private String no1GtFcbSelect;  
 private String no2GtFcbSelect;  
 private String fcbCommand;  
 private String gt1GasComb;  
 private String gt2GasComb;  
 private String gt1OilComb;  
 private String gt2OilComb;  
 private String gt1EncenderComb;  
 private String gt2EncenderComb;  
 private String actualHigherHeatingValueWet;  
 private String actualLowerHeatingValueWet;  
 private String actualHigherHeatingValueWet2;  
 private String actualLowerHeatingValueWet2;  
 private String gasVolumeTotal;  
 private String gasMassTotal;  
 private String qg55minHeatInputWithGas;  
 private String qd55minHeatInputWithOil;  
 private String gt1OperationMode;  
 private String gt2OperationMode;  
 private String spare1;  
 private String spare2;  
 private String spare3;  
 private String tipoCarga;  
 private BigDecimal cargaId;  
 private Date horaCarga;  
 private BigDecimal estatusLimpio;  
  
 public TxCrudos() {  
 }  
  
 public TxCrudos(String dia, String minuto, String fuelSelect, String hourlyLoadInUse, String line1WattHourMas5min, String line2WattHourMas5min, String line1WattHourMenos5min, String line2WattHourMenos5min, String line1WattHourMas5minBackup, String line2WattHourMas5minBackup, String line1WattHourMenos5minBackup, String line2WattHourMenos5minBackup, String gt1ActivePower, String gt2ActivePower, String stActivePower, String gt1ReactivePower, String gt2ReactivePower, String stReactivePower, String line1WattInstant, String line2WattInstant, String line1VarInstant, String line2VarInstant, String plantNetWattHourCalculationFromGross, String stgWattHour, String stgVarHour, String no1GtgWattHour, String no1GtgVarHour, String no1GtgUnitHouseLoad, String no2GtgWattHour, String no2GtgVarHour, String no2GtgUnitHouseLoad, String ambientTemperatureInUse, String ambientTemperature1, String ambientTemperature2, String relativeHumidityInUse, String relativeHumidity1, String relativeHumidity2, String ambientPressureInUse, String ambientPressure1, String ambientPressure2, String condenserVacuumInUse, String condenserVacuum1, String condenserVacuum2, String condenserVacuumMonitoring, String fuelGasHigherHeatingValue, String fuelGasLowerHeatingValue, String fuelGasHigherHeatingValue2, String fuelGasLowerHeatingValue2, String fuelOilHigherHeatingValeu, String fuelOilLowerHeatingValue, String fuelOilHigherHeatingValeu2, String fuelOilLowerHeatingValue2, String fuelOilSpecificGravity, String fuelGasFlowCurrentMasFlow, String strm1CurrentVolumeFlowRate, String fuelGasFlow5minAccumulatedValue, String oilFlowGt1Gt2, String gt1OilFlowCurrentValue, String gt2OilFlowCurrentValue, String oilFlowGt1Gt22, String gt1OilFlow5minAccumulatedValue, String gt2OilFlow5minAccumulatedValue, String overallPowerFactor, String netPowerOf5min5minAccumulatedWithGasFiring, String netPowerOf5min5minAccumulatedWithOilFiring, String loadPercentageAtGas, String loadPercentageAtOil, String ctungAtYearAverageDesignConditionGas, String ctungAtYearAverageDesignConditionOil, String correctedCtungGas, String correctedCtungOil, String fuelGasPressure, String gasTemperature, String loadDemand, String eh512RollingAverage, String ambientTemperature12RollingAverage, String relativeHumidity12RollingAverage, String ambientPressure12RollingAverage, String condenserVacuum12RollingAverage, String fuelGasFlow12RollingAverage, String fuelGasFlow12RollingAverage2, String fuelGasPressure12RollingAverage, String overallPowerFactor12RollingAverage, String fuelOilFlow12RollingAverage, String fuelOilFlow12RollingAverage2, String correctedCtung12RollingWeightedAverageGas, String correctedCtung12RollingWeightedAverageOil, String fuelGasHigherHeatingValue12RollingAverage, String fuelGasLowerHeatingValue12RollingAverage, String fuelGasHigherHeatingValue12RollingAverage2, String fuelGasLowerHeatingValue12RollingAverage2, String fuelOilHigherHeatingValue12RollingAverage, String fuelOilLowerHeatingValue12RollingAverage, String fuelOilHigherHeatingValue12RollingAverage2, String fuelOilLowerHeatingValue12RollingAverage2, String numberOfMinutesOfOperatingAtGasFiring, String numberOfMinutesOfOperatingAtOilFiring, String eh12RollingAverage, String egh12RollingAverage, String edh12RollingAverage, String nhovgOperatingHourAtNoLoadBurningGas, String nhovdOperatingHourAtNoLoadBurningOil, String gt1OperatingHour, String gt2OperatingHour, String netOutputCorrectedToSummerDesignCfe, String kcactualGas, String kcactualOil, String spare, String gt152gOn, String gt252gOn, String EA1010400kvSSCbGt1Abn, String EA1010400kvSSCbGt1Close, String EA1010400kvSSCbGt1Open, String EA1020400kvSSCbGt2Abn, String EA1020400kvSSCbGt2Close, String EA1020400kvSSCbGt2Open, String EA1030400kvSSCbStClose, String EA1030400kvSSCbStOpen, String gt1Synchronize, String gt2Synchronize, String allTariffMeterAbnormal, String gt1TransmitterWatAndVarAbnormal, String gt2TransmitterWatAndVarAbnormal, String stTransmitterWatAndVarAbnormal, String no1TempTransmitterAbnormal, String no2TempTransmitterAbnormal, String tempTransmitterSelect0No21No1, String no1HumidityTransmitterAbnormal, String no2HumidityTransmitterAbnormal, String humidityTransmitterSelect0No21No1, String no1AmbientPressureTransmitterAbnormal, String no2AmbientPressureTransmitterAbnormal, String ambientPressureTransmitterSelect0No21No1, String no1CondVacuumTransmitterAbnormal, String no2CondVacuumTransmitterAbnormal, String condVacuumTransmitterSelect0No21No1, String gt1GasSelect, String gt2GasSelect, String gt1OilSelect, String gt2OilSelect, String no1GtFcbSelect, String no2GtFcbSelect, String fcbCommand, String gt1GasComb, String gt2GasComb, String gt1OilComb, String gt2OilComb, String gt1EncenderComb, String gt2EncenderComb, String actualHigherHeatingValueWet, String actualLowerHeatingValueWet, String actualHigherHeatingValueWet2, String actualLowerHeatingValueWet2, String gasVolumeTotal, String gasMassTotal, String qg55minHeatInputWithGas, String qd55minHeatInputWithOil, String gt1OperationMode, String gt2OperationMode, String spare1, String spare2, String spare3, String tipoCarga, BigDecimal cargaId, Date horaCarga, BigDecimal estatusLimpio) {  
 this.dia = dia;  
 this.minuto = minuto;  
 this.fuelSelect = fuelSelect;  
 this.hourlyLoadInUse = hourlyLoadInUse;  
 this.line1WattHourMas5min = line1WattHourMas5min;  
 this.line2WattHourMas5min = line2WattHourMas5min;  
 this.line1WattHourMenos5min = line1WattHourMenos5min;  
 this.line2WattHourMenos5min = line2WattHourMenos5min;  
 this.line1WattHourMas5minBackup = line1WattHourMas5minBackup;  
 this.line2WattHourMas5minBackup = line2WattHourMas5minBackup;  
 this.line1WattHourMenos5minBackup = line1WattHourMenos5minBackup;  
 this.line2WattHourMenos5minBackup = line2WattHourMenos5minBackup;  
 this.gt1ActivePower = gt1ActivePower;  
 this.gt2ActivePower = gt2ActivePower;  
 this.stActivePower = stActivePower;  
 this.gt1ReactivePower = gt1ReactivePower;  
 this.gt2ReactivePower = gt2ReactivePower;  
 this.stReactivePower = stReactivePower;  
 this.line1WattInstant = line1WattInstant;  
 this.line2WattInstant = line2WattInstant;  
 this.line1VarInstant = line1VarInstant;  
 this.line2VarInstant = line2VarInstant;  
 this.plantNetWattHourCalculationFromGross = plantNetWattHourCalculationFromGross;  
 this.stgWattHour = stgWattHour;  
 this.stgVarHour = stgVarHour;  
 this.no1GtgWattHour = no1GtgWattHour;  
 this.no1GtgVarHour = no1GtgVarHour;  
 this.no1GtgUnitHouseLoad = no1GtgUnitHouseLoad;  
 this.no2GtgWattHour = no2GtgWattHour;  
 this.no2GtgVarHour = no2GtgVarHour;  
 this.no2GtgUnitHouseLoad = no2GtgUnitHouseLoad;  
 this.ambientTemperatureInUse = ambientTemperatureInUse;  
 this.ambientTemperature1 = ambientTemperature1;  
 this.ambientTemperature2 = ambientTemperature2;  
 this.relativeHumidityInUse = relativeHumidityInUse;  
 this.relativeHumidity1 = relativeHumidity1;  
 this.relativeHumidity2 = relativeHumidity2;  
 this.ambientPressureInUse = ambientPressureInUse;  
 this.ambientPressure1 = ambientPressure1;  
 this.ambientPressure2 = ambientPressure2;  
 this.condenserVacuumInUse = condenserVacuumInUse;  
 this.condenserVacuum1 = condenserVacuum1;  
 this.condenserVacuum2 = condenserVacuum2;  
 this.condenserVacuumMonitoring = condenserVacuumMonitoring;  
 this.fuelGasHigherHeatingValue = fuelGasHigherHeatingValue;  
 this.fuelGasLowerHeatingValue = fuelGasLowerHeatingValue;  
 this.fuelGasHigherHeatingValue2 = fuelGasHigherHeatingValue2;  
 this.fuelGasLowerHeatingValue2 = fuelGasLowerHeatingValue2;  
 this.fuelOilHigherHeatingValeu = fuelOilHigherHeatingValeu;  
 this.fuelOilLowerHeatingValue = fuelOilLowerHeatingValue;  
 this.fuelOilHigherHeatingValeu2 = fuelOilHigherHeatingValeu2;  
 this.fuelOilLowerHeatingValue2 = fuelOilLowerHeatingValue2;  
 this.fuelOilSpecificGravity = fuelOilSpecificGravity;  
 this.fuelGasFlowCurrentMasFlow = fuelGasFlowCurrentMasFlow;  
 this.strm1CurrentVolumeFlowRate = strm1CurrentVolumeFlowRate;  
 this.fuelGasFlow5minAccumulatedValue = fuelGasFlow5minAccumulatedValue;  
 this.oilFlowGt1Gt2 = oilFlowGt1Gt2;  
 this.gt1OilFlowCurrentValue = gt1OilFlowCurrentValue;  
 this.gt2OilFlowCurrentValue = gt2OilFlowCurrentValue;  
 this.oilFlowGt1Gt22 = oilFlowGt1Gt22;  
 this.gt1OilFlow5minAccumulatedValue = gt1OilFlow5minAccumulatedValue;  
 this.gt2OilFlow5minAccumulatedValue = gt2OilFlow5minAccumulatedValue;  
 this.overallPowerFactor = overallPowerFactor;  
 this.netPowerOf5min5minAccumulatedWithGasFiring = netPowerOf5min5minAccumulatedWithGasFiring;  
 this.netPowerOf5min5minAccumulatedWithOilFiring = netPowerOf5min5minAccumulatedWithOilFiring;  
 this.loadPercentageAtGas = loadPercentageAtGas;  
 this.loadPercentageAtOil = loadPercentageAtOil;  
 this.ctungAtYearAverageDesignConditionGas = ctungAtYearAverageDesignConditionGas;  
 this.ctungAtYearAverageDesignConditionOil = ctungAtYearAverageDesignConditionOil;  
 this.correctedCtungGas = correctedCtungGas;  
 this.correctedCtungOil = correctedCtungOil;  
 this.fuelGasPressure = fuelGasPressure;  
 this.gasTemperature = gasTemperature;  
 this.loadDemand = loadDemand;  
 this.eh512RollingAverage = eh512RollingAverage;  
 this.ambientTemperature12RollingAverage = ambientTemperature12RollingAverage;  
 this.relativeHumidity12RollingAverage = relativeHumidity12RollingAverage;  
 this.ambientPressure12RollingAverage = ambientPressure12RollingAverage;  
 this.condenserVacuum12RollingAverage = condenserVacuum12RollingAverage;  
 this.fuelGasFlow12RollingAverage = fuelGasFlow12RollingAverage;  
 this.fuelGasFlow12RollingAverage2 = fuelGasFlow12RollingAverage2;  
 this.fuelGasPressure12RollingAverage = fuelGasPressure12RollingAverage;  
 this.overallPowerFactor12RollingAverage = overallPowerFactor12RollingAverage;  
 this.fuelOilFlow12RollingAverage = fuelOilFlow12RollingAverage;  
 this.fuelOilFlow12RollingAverage2 = fuelOilFlow12RollingAverage2;  
 this.correctedCtung12RollingWeightedAverageGas = correctedCtung12RollingWeightedAverageGas;  
 this.correctedCtung12RollingWeightedAverageOil = correctedCtung12RollingWeightedAverageOil;  
 this.fuelGasHigherHeatingValue12RollingAverage = fuelGasHigherHeatingValue12RollingAverage;  
 this.fuelGasLowerHeatingValue12RollingAverage = fuelGasLowerHeatingValue12RollingAverage;  
 this.fuelGasHigherHeatingValue12RollingAverage2 = fuelGasHigherHeatingValue12RollingAverage2;  
 this.fuelGasLowerHeatingValue12RollingAverage2 = fuelGasLowerHeatingValue12RollingAverage2;  
 this.fuelOilHigherHeatingValue12RollingAverage = fuelOilHigherHeatingValue12RollingAverage;  
 this.fuelOilLowerHeatingValue12RollingAverage = fuelOilLowerHeatingValue12RollingAverage;  
 this.fuelOilHigherHeatingValue12RollingAverage2 = fuelOilHigherHeatingValue12RollingAverage2;  
 this.fuelOilLowerHeatingValue12RollingAverage2 = fuelOilLowerHeatingValue12RollingAverage2;  
 this.numberOfMinutesOfOperatingAtGasFiring = numberOfMinutesOfOperatingAtGasFiring;  
 this.numberOfMinutesOfOperatingAtOilFiring = numberOfMinutesOfOperatingAtOilFiring;  
 this.eh12RollingAverage = eh12RollingAverage;  
 this.egh12RollingAverage = egh12RollingAverage;  
 this.edh12RollingAverage = edh12RollingAverage;  
 this.nhovgOperatingHourAtNoLoadBurningGas = nhovgOperatingHourAtNoLoadBurningGas;  
 this.nhovdOperatingHourAtNoLoadBurningOil = nhovdOperatingHourAtNoLoadBurningOil;  
 this.gt1OperatingHour = gt1OperatingHour;  
 this.gt2OperatingHour = gt2OperatingHour;  
 this.netOutputCorrectedToSummerDesignCfe = netOutputCorrectedToSummerDesignCfe;  
 this.kcactualGas = kcactualGas;  
 this.kcactualOil = kcactualOil;  
 this.spare = spare;  
 this.gt152gOn = gt152gOn;  
 this.gt252gOn = gt252gOn;  
 this.EA1010400kvSSCbGt1Abn = EA1010400kvSSCbGt1Abn;  
 this.EA1010400kvSSCbGt1Close = EA1010400kvSSCbGt1Close;  
 this.EA1010400kvSSCbGt1Open = EA1010400kvSSCbGt1Open;  
 this.EA1020400kvSSCbGt2Abn = EA1020400kvSSCbGt2Abn;  
 this.EA1020400kvSSCbGt2Close = EA1020400kvSSCbGt2Close;  
 this.EA1020400kvSSCbGt2Open = EA1020400kvSSCbGt2Open;  
 this.EA1030400kvSSCbStClose = EA1030400kvSSCbStClose;  
 this.EA1030400kvSSCbStOpen = EA1030400kvSSCbStOpen;  
 this.gt1Synchronize = gt1Synchronize;  
 this.gt2Synchronize = gt2Synchronize;  
 this.allTariffMeterAbnormal = allTariffMeterAbnormal;  
 this.gt1TransmitterWatAndVarAbnormal = gt1TransmitterWatAndVarAbnormal;  
 this.gt2TransmitterWatAndVarAbnormal = gt2TransmitterWatAndVarAbnormal;  
 this.stTransmitterWatAndVarAbnormal = stTransmitterWatAndVarAbnormal;  
 this.no1TempTransmitterAbnormal = no1TempTransmitterAbnormal;  
 this.no2TempTransmitterAbnormal = no2TempTransmitterAbnormal;  
 this.tempTransmitterSelect0No21No1 = tempTransmitterSelect0No21No1;  
 this.no1HumidityTransmitterAbnormal = no1HumidityTransmitterAbnormal;  
 this.no2HumidityTransmitterAbnormal = no2HumidityTransmitterAbnormal;  
 this.humidityTransmitterSelect0No21No1 = humidityTransmitterSelect0No21No1;  
 this.no1AmbientPressureTransmitterAbnormal = no1AmbientPressureTransmitterAbnormal;  
 this.no2AmbientPressureTransmitterAbnormal = no2AmbientPressureTransmitterAbnormal;  
 this.ambientPressureTransmitterSelect0No21No1 = ambientPressureTransmitterSelect0No21No1;  
 this.no1CondVacuumTransmitterAbnormal = no1CondVacuumTransmitterAbnormal;  
 this.no2CondVacuumTransmitterAbnormal = no2CondVacuumTransmitterAbnormal;  
 this.condVacuumTransmitterSelect0No21No1 = condVacuumTransmitterSelect0No21No1;  
 this.gt1GasSelect = gt1GasSelect;  
 this.gt2GasSelect = gt2GasSelect;  
 this.gt1OilSelect = gt1OilSelect;  
 this.gt2OilSelect = gt2OilSelect;  
 this.no1GtFcbSelect = no1GtFcbSelect;  
 this.no2GtFcbSelect = no2GtFcbSelect;  
 this.fcbCommand = fcbCommand;  
 this.gt1GasComb = gt1GasComb;  
 this.gt2GasComb = gt2GasComb;  
 this.gt1OilComb = gt1OilComb;  
 this.gt2OilComb = gt2OilComb;  
 this.gt1EncenderComb = gt1EncenderComb;  
 this.gt2EncenderComb = gt2EncenderComb;  
 this.actualHigherHeatingValueWet = actualHigherHeatingValueWet;  
 this.actualLowerHeatingValueWet = actualLowerHeatingValueWet;  
 this.actualHigherHeatingValueWet2 = actualHigherHeatingValueWet2;  
 this.actualLowerHeatingValueWet2 = actualLowerHeatingValueWet2;  
 this.gasVolumeTotal = gasVolumeTotal;  
 this.gasMassTotal = gasMassTotal;  
 this.qg55minHeatInputWithGas = qg55minHeatInputWithGas;  
 this.qd55minHeatInputWithOil = qd55minHeatInputWithOil;  
 this.gt1OperationMode = gt1OperationMode;  
 this.gt2OperationMode = gt2OperationMode;  
 this.spare1 = spare1;  
 this.spare2 = spare2;  
 this.spare3 = spare3;  
 this.tipoCarga = tipoCarga;  
 this.cargaId = cargaId;  
 this.horaCarga = horaCarga;  
 this.estatusLimpio = estatusLimpio;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="CRUDOS\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getCrudosId() {  
 return this.crudosId;  
 }  
   
 public void setCrudosId(BigDecimal crudosId) {  
 this.crudosId = crudosId;  
 }  
  
   
 @Column(name="DIA", length=200)  
 public String getDia() {  
 return this.dia;  
 }  
   
 public void setDia(String dia) {  
 this.dia = dia;  
 }  
  
   
 @Column(name="MINUTO", length=200)  
 public String getMinuto() {  
 return this.minuto;  
 }  
   
 public void setMinuto(String minuto) {  
 this.minuto = minuto;  
 }  
  
   
 @Column(name="FUEL\_SELECT", length=100)  
 public String getFuelSelect() {  
 return this.fuelSelect;  
 }  
   
 public void setFuelSelect(String fuelSelect) {  
 this.fuelSelect = fuelSelect;  
 }  
  
   
 @Column(name="HOURLY\_LOAD\_IN\_USE", length=100)  
 public String getHourlyLoadInUse() {  
 return this.hourlyLoadInUse;  
 }  
   
 public void setHourlyLoadInUse(String hourlyLoadInUse) {  
 this.hourlyLoadInUse = hourlyLoadInUse;  
 }  
  
   
 @Column(name="LINE1\_WATT\_HOUR\_MAS\_5MIN", length=100)  
 public String getLine1WattHourMas5min() {  
 return this.line1WattHourMas5min;  
 }  
   
 public void setLine1WattHourMas5min(String line1WattHourMas5min) {  
 this.line1WattHourMas5min = line1WattHourMas5min;  
 }  
  
   
 @Column(name="LINE2\_WATT\_HOUR\_MAS\_5MIN", length=100)  
 public String getLine2WattHourMas5min() {  
 return this.line2WattHourMas5min;  
 }  
   
 public void setLine2WattHourMas5min(String line2WattHourMas5min) {  
 this.line2WattHourMas5min = line2WattHourMas5min;  
 }  
  
   
 @Column(name="LINE1\_WATT\_HOUR\_MENOS\_5MIN", length=100)  
 public String getLine1WattHourMenos5min() {  
 return this.line1WattHourMenos5min;  
 }  
   
 public void setLine1WattHourMenos5min(String line1WattHourMenos5min) {  
 this.line1WattHourMenos5min = line1WattHourMenos5min;  
 }  
  
   
 @Column(name="LINE2\_WATT\_HOUR\_MENOS\_5MIN", length=100)  
 public String getLine2WattHourMenos5min() {  
 return this.line2WattHourMenos5min;  
 }  
   
 public void setLine2WattHourMenos5min(String line2WattHourMenos5min) {  
 this.line2WattHourMenos5min = line2WattHourMenos5min;  
 }  
  
   
 @Column(name="LINE1\_WATT\_HOUR\_MAS\_5MIN\_BACKUP", length=100)  
 public String getLine1WattHourMas5minBackup() {  
 return this.line1WattHourMas5minBackup;  
 }  
   
 public void setLine1WattHourMas5minBackup(String line1WattHourMas5minBackup) {  
 this.line1WattHourMas5minBackup = line1WattHourMas5minBackup;  
 }  
  
   
 @Column(name="LINE2\_WATT\_HOUR\_MAS\_5MIN\_BACKUP", length=100)  
 public String getLine2WattHourMas5minBackup() {  
 return this.line2WattHourMas5minBackup;  
 }  
   
 public void setLine2WattHourMas5minBackup(String line2WattHourMas5minBackup) {  
 this.line2WattHourMas5minBackup = line2WattHourMas5minBackup;  
 }  
  
   
 @Column(name="LINE1\_WATT\_HOUR\_MENOS\_5MIN\_BACKUP", length=100)  
 public String getLine1WattHourMenos5minBackup() {  
 return this.line1WattHourMenos5minBackup;  
 }  
   
 public void setLine1WattHourMenos5minBackup(String line1WattHourMenos5minBackup) {  
 this.line1WattHourMenos5minBackup = line1WattHourMenos5minBackup;  
 }  
  
   
 @Column(name="LINE2\_WATT\_HOUR\_MENOS\_5MIN\_BACKUP", length=100)  
 public String getLine2WattHourMenos5minBackup() {  
 return this.line2WattHourMenos5minBackup;  
 }  
   
 public void setLine2WattHourMenos5minBackup(String line2WattHourMenos5minBackup) {  
 this.line2WattHourMenos5minBackup = line2WattHourMenos5minBackup;  
 }  
  
   
 @Column(name="GT1\_ACTIVE\_POWER", length=100)  
 public String getGt1ActivePower() {  
 return this.gt1ActivePower;  
 }  
   
 public void setGt1ActivePower(String gt1ActivePower) {  
 this.gt1ActivePower = gt1ActivePower;  
 }  
  
   
 @Column(name="GT2\_ACTIVE\_POWER", length=100)  
 public String getGt2ActivePower() {  
 return this.gt2ActivePower;  
 }  
   
 public void setGt2ActivePower(String gt2ActivePower) {  
 this.gt2ActivePower = gt2ActivePower;  
 }  
  
   
 @Column(name="ST\_ACTIVE\_POWER", length=100)  
 public String getStActivePower() {  
 return this.stActivePower;  
 }  
   
 public void setStActivePower(String stActivePower) {  
 this.stActivePower = stActivePower;  
 }  
  
   
 @Column(name="GT1\_REACTIVE\_POWER", length=100)  
 public String getGt1ReactivePower() {  
 return this.gt1ReactivePower;  
 }  
   
 public void setGt1ReactivePower(String gt1ReactivePower) {  
 this.gt1ReactivePower = gt1ReactivePower;  
 }  
  
   
 @Column(name="GT2\_REACTIVE\_POWER", length=100)  
 public String getGt2ReactivePower() {  
 return this.gt2ReactivePower;  
 }  
   
 public void setGt2ReactivePower(String gt2ReactivePower) {  
 this.gt2ReactivePower = gt2ReactivePower;  
 }  
  
   
 @Column(name="ST\_REACTIVE\_POWER", length=100)  
 public String getStReactivePower() {  
 return this.stReactivePower;  
 }  
   
 public void setStReactivePower(String stReactivePower) {  
 this.stReactivePower = stReactivePower;  
 }  
  
   
 @Column(name="LINE1\_WATT\_INSTANT", length=100)  
 public String getLine1WattInstant() {  
 return this.line1WattInstant;  
 }  
   
 public void setLine1WattInstant(String line1WattInstant) {  
 this.line1WattInstant = line1WattInstant;  
 }  
  
   
 @Column(name="LINE2\_WATT\_INSTANT", length=100)  
 public String getLine2WattInstant() {  
 return this.line2WattInstant;  
 }  
   
 public void setLine2WattInstant(String line2WattInstant) {  
 this.line2WattInstant = line2WattInstant;  
 }  
  
   
 @Column(name="LINE1\_VAR\_INSTANT", length=100)  
 public String getLine1VarInstant() {  
 return this.line1VarInstant;  
 }  
   
 public void setLine1VarInstant(String line1VarInstant) {  
 this.line1VarInstant = line1VarInstant;  
 }  
  
   
 @Column(name="LINE2\_VAR\_INSTANT", length=100)  
 public String getLine2VarInstant() {  
 return this.line2VarInstant;  
 }  
   
 public void setLine2VarInstant(String line2VarInstant) {  
 this.line2VarInstant = line2VarInstant;  
 }  
  
   
 @Column(name="PLANT\_NET\_WATT\_HOUR\_CALCULATION\_FROM\_GROSS", length=100)  
 public String getPlantNetWattHourCalculationFromGross() {  
 return this.plantNetWattHourCalculationFromGross;  
 }  
   
 public void setPlantNetWattHourCalculationFromGross(String plantNetWattHourCalculationFromGross) {  
 this.plantNetWattHourCalculationFromGross = plantNetWattHourCalculationFromGross;  
 }  
  
   
 @Column(name="STG\_WATT\_HOUR", length=100)  
 public String getStgWattHour() {  
 return this.stgWattHour;  
 }  
   
 public void setStgWattHour(String stgWattHour) {  
 this.stgWattHour = stgWattHour;  
 }  
  
   
 @Column(name="STG\_VAR\_HOUR", length=100)  
 public String getStgVarHour() {  
 return this.stgVarHour;  
 }  
   
 public void setStgVarHour(String stgVarHour) {  
 this.stgVarHour = stgVarHour;  
 }  
  
   
 @Column(name="NO1\_GTG\_WATT\_HOUR", length=100)  
 public String getNo1GtgWattHour() {  
 return this.no1GtgWattHour;  
 }  
   
 public void setNo1GtgWattHour(String no1GtgWattHour) {  
 this.no1GtgWattHour = no1GtgWattHour;  
 }  
  
   
 @Column(name="NO1\_GTG\_VAR\_HOUR", length=100)  
 public String getNo1GtgVarHour() {  
 return this.no1GtgVarHour;  
 }  
   
 public void setNo1GtgVarHour(String no1GtgVarHour) {  
 this.no1GtgVarHour = no1GtgVarHour;  
 }  
  
   
 @Column(name="NO1\_GTG\_UNIT\_HOUSE\_LOAD", length=100)  
 public String getNo1GtgUnitHouseLoad() {  
 return this.no1GtgUnitHouseLoad;  
 }  
   
 public void setNo1GtgUnitHouseLoad(String no1GtgUnitHouseLoad) {  
 this.no1GtgUnitHouseLoad = no1GtgUnitHouseLoad;  
 }  
  
   
 @Column(name="NO2\_GTG\_WATT\_HOUR", length=100)  
 public String getNo2GtgWattHour() {  
 return this.no2GtgWattHour;  
 }  
   
 public void setNo2GtgWattHour(String no2GtgWattHour) {  
 this.no2GtgWattHour = no2GtgWattHour;  
 }  
  
   
 @Column(name="NO2\_GTG\_VAR\_HOUR", length=100)  
 public String getNo2GtgVarHour() {  
 return this.no2GtgVarHour;  
 }  
   
 public void setNo2GtgVarHour(String no2GtgVarHour) {  
 this.no2GtgVarHour = no2GtgVarHour;  
 }  
  
   
 @Column(name="NO2\_GTG\_UNIT\_HOUSE\_LOAD", length=100)  
 public String getNo2GtgUnitHouseLoad() {  
 return this.no2GtgUnitHouseLoad;  
 }  
   
 public void setNo2GtgUnitHouseLoad(String no2GtgUnitHouseLoad) {  
 this.no2GtgUnitHouseLoad = no2GtgUnitHouseLoad;  
 }  
  
   
 @Column(name="AMBIENT\_TEMPERATURE\_IN\_USE", length=100)  
 public String getAmbientTemperatureInUse() {  
 return this.ambientTemperatureInUse;  
 }  
   
 public void setAmbientTemperatureInUse(String ambientTemperatureInUse) {  
 this.ambientTemperatureInUse = ambientTemperatureInUse;  
 }  
  
   
 @Column(name="AMBIENT\_TEMPERATURE1", length=100)  
 public String getAmbientTemperature1() {  
 return this.ambientTemperature1;  
 }  
   
 public void setAmbientTemperature1(String ambientTemperature1) {  
 this.ambientTemperature1 = ambientTemperature1;  
 }  
  
   
 @Column(name="AMBIENT\_TEMPERATURE2", length=100)  
 public String getAmbientTemperature2() {  
 return this.ambientTemperature2;  
 }  
   
 public void setAmbientTemperature2(String ambientTemperature2) {  
 this.ambientTemperature2 = ambientTemperature2;  
 }  
  
   
 @Column(name="RELATIVE\_HUMIDITY\_IN\_USE", length=100)  
 public String getRelativeHumidityInUse() {  
 return this.relativeHumidityInUse;  
 }  
   
 public void setRelativeHumidityInUse(String relativeHumidityInUse) {  
 this.relativeHumidityInUse = relativeHumidityInUse;  
 }  
  
   
 @Column(name="RELATIVE\_HUMIDITY1", length=100)  
 public String getRelativeHumidity1() {  
 return this.relativeHumidity1;  
 }  
   
 public void setRelativeHumidity1(String relativeHumidity1) {  
 this.relativeHumidity1 = relativeHumidity1;  
 }  
  
   
 @Column(name="RELATIVE\_HUMIDITY2", length=100)  
 public String getRelativeHumidity2() {  
 return this.relativeHumidity2;  
 }  
   
 public void setRelativeHumidity2(String relativeHumidity2) {  
 this.relativeHumidity2 = relativeHumidity2;  
 }  
  
   
 @Column(name="AMBIENT\_PRESSURE\_IN\_USE", length=100)  
 public String getAmbientPressureInUse() {  
 return this.ambientPressureInUse;  
 }  
   
 public void setAmbientPressureInUse(String ambientPressureInUse) {  
 this.ambientPressureInUse = ambientPressureInUse;  
 }  
  
   
 @Column(name="AMBIENT\_PRESSURE1", length=100)  
 public String getAmbientPressure1() {  
 return this.ambientPressure1;  
 }  
   
 public void setAmbientPressure1(String ambientPressure1) {  
 this.ambientPressure1 = ambientPressure1;  
 }  
  
   
 @Column(name="AMBIENT\_PRESSURE2", length=100)  
 public String getAmbientPressure2() {  
 return this.ambientPressure2;  
 }  
   
 public void setAmbientPressure2(String ambientPressure2) {  
 this.ambientPressure2 = ambientPressure2;  
 }  
  
   
 @Column(name="CONDENSER\_VACUUM\_IN\_USE", length=100)  
 public String getCondenserVacuumInUse() {  
 return this.condenserVacuumInUse;  
 }  
   
 public void setCondenserVacuumInUse(String condenserVacuumInUse) {  
 this.condenserVacuumInUse = condenserVacuumInUse;  
 }  
  
   
 @Column(name="CONDENSER\_VACUUM1", length=100)  
 public String getCondenserVacuum1() {  
 return this.condenserVacuum1;  
 }  
   
 public void setCondenserVacuum1(String condenserVacuum1) {  
 this.condenserVacuum1 = condenserVacuum1;  
 }  
  
   
 @Column(name="CONDENSER\_VACUUM2", length=100)  
 public String getCondenserVacuum2() {  
 return this.condenserVacuum2;  
 }  
   
 public void setCondenserVacuum2(String condenserVacuum2) {  
 this.condenserVacuum2 = condenserVacuum2;  
 }  
  
   
 @Column(name="CONDENSER\_VACUUM\_MONITORING", length=100)  
 public String getCondenserVacuumMonitoring() {  
 return this.condenserVacuumMonitoring;  
 }  
   
 public void setCondenserVacuumMonitoring(String condenserVacuumMonitoring) {  
 this.condenserVacuumMonitoring = condenserVacuumMonitoring;  
 }  
  
   
 @Column(name="FUEL\_GAS\_HIGHER\_HEATING\_VALUE", length=100)  
 public String getFuelGasHigherHeatingValue() {  
 return this.fuelGasHigherHeatingValue;  
 }  
   
 public void setFuelGasHigherHeatingValue(String fuelGasHigherHeatingValue) {  
 this.fuelGasHigherHeatingValue = fuelGasHigherHeatingValue;  
 }  
  
   
 @Column(name="FUEL\_GAS\_LOWER\_HEATING\_VALUE", length=100)  
 public String getFuelGasLowerHeatingValue() {  
 return this.fuelGasLowerHeatingValue;  
 }  
   
 public void setFuelGasLowerHeatingValue(String fuelGasLowerHeatingValue) {  
 this.fuelGasLowerHeatingValue = fuelGasLowerHeatingValue;  
 }  
  
   
 @Column(name="FUEL\_GAS\_HIGHER\_HEATING\_VALUE2", length=100)  
 public String getFuelGasHigherHeatingValue2() {  
 return this.fuelGasHigherHeatingValue2;  
 }  
   
 public void setFuelGasHigherHeatingValue2(String fuelGasHigherHeatingValue2) {  
 this.fuelGasHigherHeatingValue2 = fuelGasHigherHeatingValue2;  
 }  
  
   
 @Column(name="FUEL\_GAS\_LOWER\_HEATING\_VALUE2", length=100)  
 public String getFuelGasLowerHeatingValue2() {  
 return this.fuelGasLowerHeatingValue2;  
 }  
   
 public void setFuelGasLowerHeatingValue2(String fuelGasLowerHeatingValue2) {  
 this.fuelGasLowerHeatingValue2 = fuelGasLowerHeatingValue2;  
 }  
  
   
 @Column(name="FUEL\_OIL\_HIGHER\_HEATING\_VALEU", length=100)  
 public String getFuelOilHigherHeatingValeu() {  
 return this.fuelOilHigherHeatingValeu;  
 }  
   
 public void setFuelOilHigherHeatingValeu(String fuelOilHigherHeatingValeu) {  
 this.fuelOilHigherHeatingValeu = fuelOilHigherHeatingValeu;  
 }  
  
   
 @Column(name="FUEL\_OIL\_LOWER\_HEATING\_VALUE", length=100)  
 public String getFuelOilLowerHeatingValue() {  
 return this.fuelOilLowerHeatingValue;  
 }  
   
 public void setFuelOilLowerHeatingValue(String fuelOilLowerHeatingValue) {  
 this.fuelOilLowerHeatingValue = fuelOilLowerHeatingValue;  
 }  
  
   
 @Column(name="FUEL\_OIL\_HIGHER\_HEATING\_VALEU2", length=100)  
 public String getFuelOilHigherHeatingValeu2() {  
 return this.fuelOilHigherHeatingValeu2;  
 }  
   
 public void setFuelOilHigherHeatingValeu2(String fuelOilHigherHeatingValeu2) {  
 this.fuelOilHigherHeatingValeu2 = fuelOilHigherHeatingValeu2;  
 }  
  
   
 @Column(name="FUEL\_OIL\_LOWER\_HEATING\_VALUE2", length=100)  
 public String getFuelOilLowerHeatingValue2() {  
 return this.fuelOilLowerHeatingValue2;  
 }  
   
 public void setFuelOilLowerHeatingValue2(String fuelOilLowerHeatingValue2) {  
 this.fuelOilLowerHeatingValue2 = fuelOilLowerHeatingValue2;  
 }  
  
   
 @Column(name="FUEL\_OIL\_SPECIFIC\_GRAVITY", length=100)  
 public String getFuelOilSpecificGravity() {  
 return this.fuelOilSpecificGravity;  
 }  
   
 public void setFuelOilSpecificGravity(String fuelOilSpecificGravity) {  
 this.fuelOilSpecificGravity = fuelOilSpecificGravity;  
 }  
  
   
 @Column(name="FUEL\_GAS\_FLOW\_CURRENT\_MAS\_FLOW", length=100)  
 public String getFuelGasFlowCurrentMasFlow() {  
 return this.fuelGasFlowCurrentMasFlow;  
 }  
   
 public void setFuelGasFlowCurrentMasFlow(String fuelGasFlowCurrentMasFlow) {  
 this.fuelGasFlowCurrentMasFlow = fuelGasFlowCurrentMasFlow;  
 }  
  
   
 @Column(name="STRM1\_CURRENT\_VOLUME\_FLOW\_RATE", length=100)  
 public String getStrm1CurrentVolumeFlowRate() {  
 return this.strm1CurrentVolumeFlowRate;  
 }  
   
 public void setStrm1CurrentVolumeFlowRate(String strm1CurrentVolumeFlowRate) {  
 this.strm1CurrentVolumeFlowRate = strm1CurrentVolumeFlowRate;  
 }  
  
   
 @Column(name="FUEL\_GAS\_FLOW\_5MIN\_ACCUMULATED\_VALUE", length=100)  
 public String getFuelGasFlow5minAccumulatedValue() {  
 return this.fuelGasFlow5minAccumulatedValue;  
 }  
   
 public void setFuelGasFlow5minAccumulatedValue(String fuelGasFlow5minAccumulatedValue) {  
 this.fuelGasFlow5minAccumulatedValue = fuelGasFlow5minAccumulatedValue;  
 }  
  
   
 @Column(name="OIL\_FLOW\_GT1\_GT2", length=100)  
 public String getOilFlowGt1Gt2() {  
 return this.oilFlowGt1Gt2;  
 }  
   
 public void setOilFlowGt1Gt2(String oilFlowGt1Gt2) {  
 this.oilFlowGt1Gt2 = oilFlowGt1Gt2;  
 }  
  
   
 @Column(name="GT1\_OIL\_FLOW\_CURRENT\_VALUE", length=100)  
 public String getGt1OilFlowCurrentValue() {  
 return this.gt1OilFlowCurrentValue;  
 }  
   
 public void setGt1OilFlowCurrentValue(String gt1OilFlowCurrentValue) {  
 this.gt1OilFlowCurrentValue = gt1OilFlowCurrentValue;  
 }  
  
   
 @Column(name="GT2\_OIL\_FLOW\_CURRENT\_VALUE", length=100)  
 public String getGt2OilFlowCurrentValue() {  
 return this.gt2OilFlowCurrentValue;  
 }  
   
 public void setGt2OilFlowCurrentValue(String gt2OilFlowCurrentValue) {  
 this.gt2OilFlowCurrentValue = gt2OilFlowCurrentValue;  
 }  
  
   
 @Column(name="OIL\_FLOW\_GT1\_GT22", length=100)  
 public String getOilFlowGt1Gt22() {  
 return this.oilFlowGt1Gt22;  
 }  
   
 public void setOilFlowGt1Gt22(String oilFlowGt1Gt22) {  
 this.oilFlowGt1Gt22 = oilFlowGt1Gt22;  
 }  
  
   
 @Column(name="GT1\_OIL\_FLOW\_5MIN\_ACCUMULATED\_VALUE", length=100)  
 public String getGt1OilFlow5minAccumulatedValue() {  
 return this.gt1OilFlow5minAccumulatedValue;  
 }  
   
 public void setGt1OilFlow5minAccumulatedValue(String gt1OilFlow5minAccumulatedValue) {  
 this.gt1OilFlow5minAccumulatedValue = gt1OilFlow5minAccumulatedValue;  
 }  
  
   
 @Column(name="GT2\_OIL\_FLOW\_5MIN\_ACCUMULATED\_VALUE", length=100)  
 public String getGt2OilFlow5minAccumulatedValue() {  
 return this.gt2OilFlow5minAccumulatedValue;  
 }  
   
 public void setGt2OilFlow5minAccumulatedValue(String gt2OilFlow5minAccumulatedValue) {  
 this.gt2OilFlow5minAccumulatedValue = gt2OilFlow5minAccumulatedValue;  
 }  
  
   
 @Column(name="OVERALL\_POWER\_FACTOR", length=100)  
 public String getOverallPowerFactor() {  
 return this.overallPowerFactor;  
 }  
   
 public void setOverallPowerFactor(String overallPowerFactor) {  
 this.overallPowerFactor = overallPowerFactor;  
 }  
  
   
 @Column(name="NET\_POWER\_OF\_5MIN\_5MIN\_ACCUMULATED\_WITH\_GAS\_FIRING", length=100)  
 public String getNetPowerOf5min5minAccumulatedWithGasFiring() {  
 return this.netPowerOf5min5minAccumulatedWithGasFiring;  
 }  
   
 public void setNetPowerOf5min5minAccumulatedWithGasFiring(String netPowerOf5min5minAccumulatedWithGasFiring) {  
 this.netPowerOf5min5minAccumulatedWithGasFiring = netPowerOf5min5minAccumulatedWithGasFiring;  
 }  
  
   
 @Column(name="NET\_POWER\_OF\_5MIN\_5MIN\_ACCUMULATED\_WITH\_OIL\_FIRING", length=100)  
 public String getNetPowerOf5min5minAccumulatedWithOilFiring() {  
 return this.netPowerOf5min5minAccumulatedWithOilFiring;  
 }  
   
 public void setNetPowerOf5min5minAccumulatedWithOilFiring(String netPowerOf5min5minAccumulatedWithOilFiring) {  
 this.netPowerOf5min5minAccumulatedWithOilFiring = netPowerOf5min5minAccumulatedWithOilFiring;  
 }  
  
   
 @Column(name="LOAD\_PERCENTAGE\_AT\_GAS", length=100)  
 public String getLoadPercentageAtGas() {  
 return this.loadPercentageAtGas;  
 }  
   
 public void setLoadPercentageAtGas(String loadPercentageAtGas) {  
 this.loadPercentageAtGas = loadPercentageAtGas;  
 }  
  
   
 @Column(name="LOAD\_PERCENTAGE\_AT\_OIL", length=100)  
 public String getLoadPercentageAtOil() {  
 return this.loadPercentageAtOil;  
 }  
   
 public void setLoadPercentageAtOil(String loadPercentageAtOil) {  
 this.loadPercentageAtOil = loadPercentageAtOil;  
 }  
  
   
 @Column(name="CTUNG\_AT\_YEAR\_AVERAGE\_DESIGN\_CONDITION\_GAS", length=100)  
 public String getCtungAtYearAverageDesignConditionGas() {  
 return this.ctungAtYearAverageDesignConditionGas;  
 }  
   
 public void setCtungAtYearAverageDesignConditionGas(String ctungAtYearAverageDesignConditionGas) {  
 this.ctungAtYearAverageDesignConditionGas = ctungAtYearAverageDesignConditionGas;  
 }  
  
   
 @Column(name="CTUNG\_AT\_YEAR\_AVERAGE\_DESIGN\_CONDITION\_OIL", length=100)  
 public String getCtungAtYearAverageDesignConditionOil() {  
 return this.ctungAtYearAverageDesignConditionOil;  
 }  
   
 public void setCtungAtYearAverageDesignConditionOil(String ctungAtYearAverageDesignConditionOil) {  
 this.ctungAtYearAverageDesignConditionOil = ctungAtYearAverageDesignConditionOil;  
 }  
  
   
 @Column(name="CORRECTED\_CTUNG\_GAS", length=100)  
 public String getCorrectedCtungGas() {  
 return this.correctedCtungGas;  
 }  
   
 public void setCorrectedCtungGas(String correctedCtungGas) {  
 this.correctedCtungGas = correctedCtungGas;  
 }  
  
   
 @Column(name="CORRECTED\_CTUNG\_OIL", length=100)  
 public String getCorrectedCtungOil() {  
 return this.correctedCtungOil;  
 }  
   
 public void setCorrectedCtungOil(String correctedCtungOil) {  
 this.correctedCtungOil = correctedCtungOil;  
 }  
  
   
 @Column(name="FUEL\_GAS\_PRESSURE", length=100)  
 public String getFuelGasPressure() {  
 return this.fuelGasPressure;  
 }  
   
 public void setFuelGasPressure(String fuelGasPressure) {  
 this.fuelGasPressure = fuelGasPressure;  
 }  
  
   
 @Column(name="GAS\_TEMPERATURE", length=100)  
 public String getGasTemperature() {  
 return this.gasTemperature;  
 }  
   
 public void setGasTemperature(String gasTemperature) {  
 this.gasTemperature = gasTemperature;  
 }  
  
   
 @Column(name="LOAD\_DEMAND", length=100)  
 public String getLoadDemand() {  
 return this.loadDemand;  
 }  
   
 public void setLoadDemand(String loadDemand) {  
 this.loadDemand = loadDemand;  
 }  
  
   
 @Column(name="EH5\_12\_ROLLING\_AVERAGE", length=100)  
 public String getEh512RollingAverage() {  
 return this.eh512RollingAverage;  
 }  
   
 public void setEh512RollingAverage(String eh512RollingAverage) {  
 this.eh512RollingAverage = eh512RollingAverage;  
 }  
  
   
 @Column(name="AMBIENT\_TEMPERATURE\_12\_ROLLING\_AVERAGE", length=100)  
 public String getAmbientTemperature12RollingAverage() {  
 return this.ambientTemperature12RollingAverage;  
 }  
   
 public void setAmbientTemperature12RollingAverage(String ambientTemperature12RollingAverage) {  
 this.ambientTemperature12RollingAverage = ambientTemperature12RollingAverage;  
 }  
  
   
 @Column(name="RELATIVE\_HUMIDITY\_12\_ROLLING\_AVERAGE", length=100)  
 public String getRelativeHumidity12RollingAverage() {  
 return this.relativeHumidity12RollingAverage;  
 }  
   
 public void setRelativeHumidity12RollingAverage(String relativeHumidity12RollingAverage) {  
 this.relativeHumidity12RollingAverage = relativeHumidity12RollingAverage;  
 }  
  
   
 @Column(name="AMBIENT\_PRESSURE\_12\_ROLLING\_AVERAGE", length=100)  
 public String getAmbientPressure12RollingAverage() {  
 return this.ambientPressure12RollingAverage;  
 }  
   
 public void setAmbientPressure12RollingAverage(String ambientPressure12RollingAverage) {  
 this.ambientPressure12RollingAverage = ambientPressure12RollingAverage;  
 }  
  
   
 @Column(name="CONDENSER\_VACUUM\_12\_ROLLING\_AVERAGE", length=100)  
 public String getCondenserVacuum12RollingAverage() {  
 return this.condenserVacuum12RollingAverage;  
 }  
   
 public void setCondenserVacuum12RollingAverage(String condenserVacuum12RollingAverage) {  
 this.condenserVacuum12RollingAverage = condenserVacuum12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_GAS\_FLOW\_12\_ROLLING\_AVERAGE", length=100)  
 public String getFuelGasFlow12RollingAverage() {  
 return this.fuelGasFlow12RollingAverage;  
 }  
   
 public void setFuelGasFlow12RollingAverage(String fuelGasFlow12RollingAverage) {  
 this.fuelGasFlow12RollingAverage = fuelGasFlow12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_GAS\_FLOW\_12\_ROLLING\_AVERAGE2", length=100)  
 public String getFuelGasFlow12RollingAverage2() {  
 return this.fuelGasFlow12RollingAverage2;  
 }  
   
 public void setFuelGasFlow12RollingAverage2(String fuelGasFlow12RollingAverage2) {  
 this.fuelGasFlow12RollingAverage2 = fuelGasFlow12RollingAverage2;  
 }  
  
   
 @Column(name="FUEL\_GAS\_PRESSURE\_12\_ROLLING\_AVERAGE", length=100)  
 public String getFuelGasPressure12RollingAverage() {  
 return this.fuelGasPressure12RollingAverage;  
 }  
   
 public void setFuelGasPressure12RollingAverage(String fuelGasPressure12RollingAverage) {  
 this.fuelGasPressure12RollingAverage = fuelGasPressure12RollingAverage;  
 }  
  
   
 @Column(name="OVERALL\_POWER\_FACTOR\_12\_ROLLING\_AVERAGE", length=100)  
 public String getOverallPowerFactor12RollingAverage() {  
 return this.overallPowerFactor12RollingAverage;  
 }  
   
 public void setOverallPowerFactor12RollingAverage(String overallPowerFactor12RollingAverage) {  
 this.overallPowerFactor12RollingAverage = overallPowerFactor12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_OIL\_FLOW\_12\_ROLLING\_AVERAGE", length=100)  
 public String getFuelOilFlow12RollingAverage() {  
 return this.fuelOilFlow12RollingAverage;  
 }  
   
 public void setFuelOilFlow12RollingAverage(String fuelOilFlow12RollingAverage) {  
 this.fuelOilFlow12RollingAverage = fuelOilFlow12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_OIL\_FLOW\_12\_ROLLING\_AVERAGE2", length=100)  
 public String getFuelOilFlow12RollingAverage2() {  
 return this.fuelOilFlow12RollingAverage2;  
 }  
   
 public void setFuelOilFlow12RollingAverage2(String fuelOilFlow12RollingAverage2) {  
 this.fuelOilFlow12RollingAverage2 = fuelOilFlow12RollingAverage2;  
 }  
  
   
 @Column(name="CORRECTED\_CTUNG\_12\_ROLLING\_WEIGHTED\_AVERAGE\_GAS", length=100)  
 public String getCorrectedCtung12RollingWeightedAverageGas() {  
 return this.correctedCtung12RollingWeightedAverageGas;  
 }  
   
 public void setCorrectedCtung12RollingWeightedAverageGas(String correctedCtung12RollingWeightedAverageGas) {  
 this.correctedCtung12RollingWeightedAverageGas = correctedCtung12RollingWeightedAverageGas;  
 }  
  
   
 @Column(name="CORRECTED\_CTUNG\_12\_ROLLING\_WEIGHTED\_AVERAGE\_OIL", length=100)  
 public String getCorrectedCtung12RollingWeightedAverageOil() {  
 return this.correctedCtung12RollingWeightedAverageOil;  
 }  
   
 public void setCorrectedCtung12RollingWeightedAverageOil(String correctedCtung12RollingWeightedAverageOil) {  
 this.correctedCtung12RollingWeightedAverageOil = correctedCtung12RollingWeightedAverageOil;  
 }  
  
   
 @Column(name="FUEL\_GAS\_HIGHER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE", length=100)  
 public String getFuelGasHigherHeatingValue12RollingAverage() {  
 return this.fuelGasHigherHeatingValue12RollingAverage;  
 }  
   
 public void setFuelGasHigherHeatingValue12RollingAverage(String fuelGasHigherHeatingValue12RollingAverage) {  
 this.fuelGasHigherHeatingValue12RollingAverage = fuelGasHigherHeatingValue12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_GAS\_LOWER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE", length=100)  
 public String getFuelGasLowerHeatingValue12RollingAverage() {  
 return this.fuelGasLowerHeatingValue12RollingAverage;  
 }  
   
 public void setFuelGasLowerHeatingValue12RollingAverage(String fuelGasLowerHeatingValue12RollingAverage) {  
 this.fuelGasLowerHeatingValue12RollingAverage = fuelGasLowerHeatingValue12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_GAS\_HIGHER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE2", length=100)  
 public String getFuelGasHigherHeatingValue12RollingAverage2() {  
 return this.fuelGasHigherHeatingValue12RollingAverage2;  
 }  
   
 public void setFuelGasHigherHeatingValue12RollingAverage2(String fuelGasHigherHeatingValue12RollingAverage2) {  
 this.fuelGasHigherHeatingValue12RollingAverage2 = fuelGasHigherHeatingValue12RollingAverage2;  
 }  
  
   
 @Column(name="FUEL\_GAS\_LOWER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE2", length=100)  
 public String getFuelGasLowerHeatingValue12RollingAverage2() {  
 return this.fuelGasLowerHeatingValue12RollingAverage2;  
 }  
   
 public void setFuelGasLowerHeatingValue12RollingAverage2(String fuelGasLowerHeatingValue12RollingAverage2) {  
 this.fuelGasLowerHeatingValue12RollingAverage2 = fuelGasLowerHeatingValue12RollingAverage2;  
 }  
  
   
 @Column(name="FUEL\_OIL\_HIGHER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE", length=100)  
 public String getFuelOilHigherHeatingValue12RollingAverage() {  
 return this.fuelOilHigherHeatingValue12RollingAverage;  
 }  
   
 public void setFuelOilHigherHeatingValue12RollingAverage(String fuelOilHigherHeatingValue12RollingAverage) {  
 this.fuelOilHigherHeatingValue12RollingAverage = fuelOilHigherHeatingValue12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_OIL\_LOWER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE", length=100)  
 public String getFuelOilLowerHeatingValue12RollingAverage() {  
 return this.fuelOilLowerHeatingValue12RollingAverage;  
 }  
   
 public void setFuelOilLowerHeatingValue12RollingAverage(String fuelOilLowerHeatingValue12RollingAverage) {  
 this.fuelOilLowerHeatingValue12RollingAverage = fuelOilLowerHeatingValue12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_OIL\_HIGHER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE2", length=100)  
 public String getFuelOilHigherHeatingValue12RollingAverage2() {  
 return this.fuelOilHigherHeatingValue12RollingAverage2;  
 }  
   
 public void setFuelOilHigherHeatingValue12RollingAverage2(String fuelOilHigherHeatingValue12RollingAverage2) {  
 this.fuelOilHigherHeatingValue12RollingAverage2 = fuelOilHigherHeatingValue12RollingAverage2;  
 }  
  
   
 @Column(name="FUEL\_OIL\_LOWER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE2", length=100)  
 public String getFuelOilLowerHeatingValue12RollingAverage2() {  
 return this.fuelOilLowerHeatingValue12RollingAverage2;  
 }  
   
 public void setFuelOilLowerHeatingValue12RollingAverage2(String fuelOilLowerHeatingValue12RollingAverage2) {  
 this.fuelOilLowerHeatingValue12RollingAverage2 = fuelOilLowerHeatingValue12RollingAverage2;  
 }  
  
   
 @Column(name="NUMBER\_OF\_MINUTES\_OF\_OPERATING\_AT\_GAS\_FIRING", length=100)  
 public String getNumberOfMinutesOfOperatingAtGasFiring() {  
 return this.numberOfMinutesOfOperatingAtGasFiring;  
 }  
   
 public void setNumberOfMinutesOfOperatingAtGasFiring(String numberOfMinutesOfOperatingAtGasFiring) {  
 this.numberOfMinutesOfOperatingAtGasFiring = numberOfMinutesOfOperatingAtGasFiring;  
 }  
  
   
 @Column(name="NUMBER\_OF\_MINUTES\_OF\_OPERATING\_AT\_OIL\_FIRING", length=100)  
 public String getNumberOfMinutesOfOperatingAtOilFiring() {  
 return this.numberOfMinutesOfOperatingAtOilFiring;  
 }  
   
 public void setNumberOfMinutesOfOperatingAtOilFiring(String numberOfMinutesOfOperatingAtOilFiring) {  
 this.numberOfMinutesOfOperatingAtOilFiring = numberOfMinutesOfOperatingAtOilFiring;  
 }  
  
   
 @Column(name="EH\_12\_ROLLING\_AVERAGE", length=100)  
 public String getEh12RollingAverage() {  
 return this.eh12RollingAverage;  
 }  
   
 public void setEh12RollingAverage(String eh12RollingAverage) {  
 this.eh12RollingAverage = eh12RollingAverage;  
 }  
  
   
 @Column(name="EGH\_12\_ROLLING\_AVERAGE", length=100)  
 public String getEgh12RollingAverage() {  
 return this.egh12RollingAverage;  
 }  
   
 public void setEgh12RollingAverage(String egh12RollingAverage) {  
 this.egh12RollingAverage = egh12RollingAverage;  
 }  
  
   
 @Column(name="EDH\_12\_ROLLING\_AVERAGE", length=100)  
 public String getEdh12RollingAverage() {  
 return this.edh12RollingAverage;  
 }  
   
 public void setEdh12RollingAverage(String edh12RollingAverage) {  
 this.edh12RollingAverage = edh12RollingAverage;  
 }  
  
   
 @Column(name="NHOVG\_OPERATING\_HOUR\_AT\_NO\_LOAD\_BURNING\_GAS", length=100)  
 public String getNhovgOperatingHourAtNoLoadBurningGas() {  
 return this.nhovgOperatingHourAtNoLoadBurningGas;  
 }  
   
 public void setNhovgOperatingHourAtNoLoadBurningGas(String nhovgOperatingHourAtNoLoadBurningGas) {  
 this.nhovgOperatingHourAtNoLoadBurningGas = nhovgOperatingHourAtNoLoadBurningGas;  
 }  
  
   
 @Column(name="NHOVD\_OPERATING\_HOUR\_AT\_NO\_LOAD\_BURNING\_OIL", length=100)  
 public String getNhovdOperatingHourAtNoLoadBurningOil() {  
 return this.nhovdOperatingHourAtNoLoadBurningOil;  
 }  
   
 public void setNhovdOperatingHourAtNoLoadBurningOil(String nhovdOperatingHourAtNoLoadBurningOil) {  
 this.nhovdOperatingHourAtNoLoadBurningOil = nhovdOperatingHourAtNoLoadBurningOil;  
 }  
  
   
 @Column(name="GT1\_OPERATING\_HOUR", length=100)  
 public String getGt1OperatingHour() {  
 return this.gt1OperatingHour;  
 }  
   
 public void setGt1OperatingHour(String gt1OperatingHour) {  
 this.gt1OperatingHour = gt1OperatingHour;  
 }  
  
   
 @Column(name="GT2\_OPERATING\_HOUR", length=100)  
 public String getGt2OperatingHour() {  
 return this.gt2OperatingHour;  
 }  
   
 public void setGt2OperatingHour(String gt2OperatingHour) {  
 this.gt2OperatingHour = gt2OperatingHour;  
 }  
  
   
 @Column(name="NET\_OUTPUT\_CORRECTED\_TO\_SUMMER\_DESIGN\_CFE", length=100)  
 public String getNetOutputCorrectedToSummerDesignCfe() {  
 return this.netOutputCorrectedToSummerDesignCfe;  
 }  
   
 public void setNetOutputCorrectedToSummerDesignCfe(String netOutputCorrectedToSummerDesignCfe) {  
 this.netOutputCorrectedToSummerDesignCfe = netOutputCorrectedToSummerDesignCfe;  
 }  
  
   
 @Column(name="KCACTUAL\_GAS", length=100)  
 public String getKcactualGas() {  
 return this.kcactualGas;  
 }  
   
 public void setKcactualGas(String kcactualGas) {  
 this.kcactualGas = kcactualGas;  
 }  
  
   
 @Column(name="KCACTUAL\_OIL", length=100)  
 public String getKcactualOil() {  
 return this.kcactualOil;  
 }  
   
 public void setKcactualOil(String kcactualOil) {  
 this.kcactualOil = kcactualOil;  
 }  
  
   
 @Column(name="SPARE", length=100)  
 public String getSpare() {  
 return this.spare;  
 }  
   
 public void setSpare(String spare) {  
 this.spare = spare;  
 }  
  
   
 @Column(name="GT1\_52G\_ON", length=100)  
 public String getGt152gOn() {  
 return this.gt152gOn;  
 }  
   
 public void setGt152gOn(String gt152gOn) {  
 this.gt152gOn = gt152gOn;  
 }  
  
   
 @Column(name="GT2\_52G\_ON", length=100)  
 public String getGt252gOn() {  
 return this.gt252gOn;  
 }  
   
 public void setGt252gOn(String gt252gOn) {  
 this.gt252gOn = gt252gOn;  
 }  
  
   
 @Column(name="E\_A1010\_400KV\_S\_S\_CB\_GT1\_ABN", length=100)  
 public String getEA1010400kvSSCbGt1Abn() {  
 return this.EA1010400kvSSCbGt1Abn;  
 }  
   
 public void setEA1010400kvSSCbGt1Abn(String EA1010400kvSSCbGt1Abn) {  
 this.EA1010400kvSSCbGt1Abn = EA1010400kvSSCbGt1Abn;  
 }  
  
   
 @Column(name="E\_A1010\_400KV\_S\_S\_CB\_GT1\_CLOSE", length=100)  
 public String getEA1010400kvSSCbGt1Close() {  
 return this.EA1010400kvSSCbGt1Close;  
 }  
   
 public void setEA1010400kvSSCbGt1Close(String EA1010400kvSSCbGt1Close) {  
 this.EA1010400kvSSCbGt1Close = EA1010400kvSSCbGt1Close;  
 }  
  
   
 @Column(name="E\_A1010\_400KV\_S\_S\_CB\_GT1\_OPEN", length=100)  
 public String getEA1010400kvSSCbGt1Open() {  
 return this.EA1010400kvSSCbGt1Open;  
 }  
   
 public void setEA1010400kvSSCbGt1Open(String EA1010400kvSSCbGt1Open) {  
 this.EA1010400kvSSCbGt1Open = EA1010400kvSSCbGt1Open;  
 }  
  
   
 @Column(name="E\_A1020\_400KV\_S\_S\_CB\_GT2\_ABN", length=100)  
 public String getEA1020400kvSSCbGt2Abn() {  
 return this.EA1020400kvSSCbGt2Abn;  
 }  
   
 public void setEA1020400kvSSCbGt2Abn(String EA1020400kvSSCbGt2Abn) {  
 this.EA1020400kvSSCbGt2Abn = EA1020400kvSSCbGt2Abn;  
 }  
  
   
 @Column(name="E\_A1020\_400KV\_S\_S\_CB\_GT2\_CLOSE", length=100)  
 public String getEA1020400kvSSCbGt2Close() {  
 return this.EA1020400kvSSCbGt2Close;  
 }  
   
 public void setEA1020400kvSSCbGt2Close(String EA1020400kvSSCbGt2Close) {  
 this.EA1020400kvSSCbGt2Close = EA1020400kvSSCbGt2Close;  
 }  
  
   
 @Column(name="E\_A1020\_400KV\_S\_S\_CB\_GT2\_OPEN", length=100)  
 public String getEA1020400kvSSCbGt2Open() {  
 return this.EA1020400kvSSCbGt2Open;  
 }  
   
 public void setEA1020400kvSSCbGt2Open(String EA1020400kvSSCbGt2Open) {  
 this.EA1020400kvSSCbGt2Open = EA1020400kvSSCbGt2Open;  
 }  
  
   
 @Column(name="E\_A1030\_400KV\_S\_S\_CB\_ST\_CLOSE", length=100)  
 public String getEA1030400kvSSCbStClose() {  
 return this.EA1030400kvSSCbStClose;  
 }  
   
 public void setEA1030400kvSSCbStClose(String EA1030400kvSSCbStClose) {  
 this.EA1030400kvSSCbStClose = EA1030400kvSSCbStClose;  
 }  
  
   
 @Column(name="E\_A1030\_400KV\_S\_S\_CB\_ST\_OPEN", length=100)  
 public String getEA1030400kvSSCbStOpen() {  
 return this.EA1030400kvSSCbStOpen;  
 }  
   
 public void setEA1030400kvSSCbStOpen(String EA1030400kvSSCbStOpen) {  
 this.EA1030400kvSSCbStOpen = EA1030400kvSSCbStOpen;  
 }  
  
   
 @Column(name="GT1\_SYNCHRONIZE", length=100)  
 public String getGt1Synchronize() {  
 return this.gt1Synchronize;  
 }  
   
 public void setGt1Synchronize(String gt1Synchronize) {  
 this.gt1Synchronize = gt1Synchronize;  
 }  
  
   
 @Column(name="GT2\_SYNCHRONIZE", length=100)  
 public String getGt2Synchronize() {  
 return this.gt2Synchronize;  
 }  
   
 public void setGt2Synchronize(String gt2Synchronize) {  
 this.gt2Synchronize = gt2Synchronize;  
 }  
  
   
 @Column(name="ALL\_TARIFF\_METER\_ABNORMAL", length=100)  
 public String getAllTariffMeterAbnormal() {  
 return this.allTariffMeterAbnormal;  
 }  
   
 public void setAllTariffMeterAbnormal(String allTariffMeterAbnormal) {  
 this.allTariffMeterAbnormal = allTariffMeterAbnormal;  
 }  
  
   
 @Column(name="GT1\_TRANSMITTER\_WAT\_AND\_VAR\_ABNORMAL", length=100)  
 public String getGt1TransmitterWatAndVarAbnormal() {  
 return this.gt1TransmitterWatAndVarAbnormal;  
 }  
   
 public void setGt1TransmitterWatAndVarAbnormal(String gt1TransmitterWatAndVarAbnormal) {  
 this.gt1TransmitterWatAndVarAbnormal = gt1TransmitterWatAndVarAbnormal;  
 }  
  
   
 @Column(name="GT2\_TRANSMITTER\_WAT\_AND\_VAR\_ABNORMAL", length=100)  
 public String getGt2TransmitterWatAndVarAbnormal() {  
 return this.gt2TransmitterWatAndVarAbnormal;  
 }  
   
 public void setGt2TransmitterWatAndVarAbnormal(String gt2TransmitterWatAndVarAbnormal) {  
 this.gt2TransmitterWatAndVarAbnormal = gt2TransmitterWatAndVarAbnormal;  
 }  
  
   
 @Column(name="ST\_TRANSMITTER\_WAT\_AND\_VAR\_ABNORMAL", length=100)  
 public String getStTransmitterWatAndVarAbnormal() {  
 return this.stTransmitterWatAndVarAbnormal;  
 }  
   
 public void setStTransmitterWatAndVarAbnormal(String stTransmitterWatAndVarAbnormal) {  
 this.stTransmitterWatAndVarAbnormal = stTransmitterWatAndVarAbnormal;  
 }  
  
   
 @Column(name="NO1\_TEMP\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo1TempTransmitterAbnormal() {  
 return this.no1TempTransmitterAbnormal;  
 }  
   
 public void setNo1TempTransmitterAbnormal(String no1TempTransmitterAbnormal) {  
 this.no1TempTransmitterAbnormal = no1TempTransmitterAbnormal;  
 }  
  
   
 @Column(name="NO2\_TEMP\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo2TempTransmitterAbnormal() {  
 return this.no2TempTransmitterAbnormal;  
 }  
   
 public void setNo2TempTransmitterAbnormal(String no2TempTransmitterAbnormal) {  
 this.no2TempTransmitterAbnormal = no2TempTransmitterAbnormal;  
 }  
  
   
 @Column(name="TEMP\_TRANSMITTER\_SELECT\_0\_NO2\_1\_NO1", length=100)  
 public String getTempTransmitterSelect0No21No1() {  
 return this.tempTransmitterSelect0No21No1;  
 }  
   
 public void setTempTransmitterSelect0No21No1(String tempTransmitterSelect0No21No1) {  
 this.tempTransmitterSelect0No21No1 = tempTransmitterSelect0No21No1;  
 }  
  
   
 @Column(name="NO1\_HUMIDITY\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo1HumidityTransmitterAbnormal() {  
 return this.no1HumidityTransmitterAbnormal;  
 }  
   
 public void setNo1HumidityTransmitterAbnormal(String no1HumidityTransmitterAbnormal) {  
 this.no1HumidityTransmitterAbnormal = no1HumidityTransmitterAbnormal;  
 }  
  
   
 @Column(name="NO2\_HUMIDITY\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo2HumidityTransmitterAbnormal() {  
 return this.no2HumidityTransmitterAbnormal;  
 }  
   
 public void setNo2HumidityTransmitterAbnormal(String no2HumidityTransmitterAbnormal) {  
 this.no2HumidityTransmitterAbnormal = no2HumidityTransmitterAbnormal;  
 }  
  
   
 @Column(name="HUMIDITY\_TRANSMITTER\_SELECT\_0\_NO2\_1\_NO1", length=100)  
 public String getHumidityTransmitterSelect0No21No1() {  
 return this.humidityTransmitterSelect0No21No1;  
 }  
   
 public void setHumidityTransmitterSelect0No21No1(String humidityTransmitterSelect0No21No1) {  
 this.humidityTransmitterSelect0No21No1 = humidityTransmitterSelect0No21No1;  
 }  
  
   
 @Column(name="NO1\_AMBIENT\_PRESSURE\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo1AmbientPressureTransmitterAbnormal() {  
 return this.no1AmbientPressureTransmitterAbnormal;  
 }  
   
 public void setNo1AmbientPressureTransmitterAbnormal(String no1AmbientPressureTransmitterAbnormal) {  
 this.no1AmbientPressureTransmitterAbnormal = no1AmbientPressureTransmitterAbnormal;  
 }  
  
   
 @Column(name="NO2\_AMBIENT\_PRESSURE\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo2AmbientPressureTransmitterAbnormal() {  
 return this.no2AmbientPressureTransmitterAbnormal;  
 }  
   
 public void setNo2AmbientPressureTransmitterAbnormal(String no2AmbientPressureTransmitterAbnormal) {  
 this.no2AmbientPressureTransmitterAbnormal = no2AmbientPressureTransmitterAbnormal;  
 }  
  
   
 @Column(name="AMBIENT\_PRESSURE\_TRANSMITTER\_SELECT\_0\_NO2\_1\_NO1", length=100)  
 public String getAmbientPressureTransmitterSelect0No21No1() {  
 return this.ambientPressureTransmitterSelect0No21No1;  
 }  
   
 public void setAmbientPressureTransmitterSelect0No21No1(String ambientPressureTransmitterSelect0No21No1) {  
 this.ambientPressureTransmitterSelect0No21No1 = ambientPressureTransmitterSelect0No21No1;  
 }  
  
   
 @Column(name="NO1\_COND\_VACUUM\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo1CondVacuumTransmitterAbnormal() {  
 return this.no1CondVacuumTransmitterAbnormal;  
 }  
   
 public void setNo1CondVacuumTransmitterAbnormal(String no1CondVacuumTransmitterAbnormal) {  
 this.no1CondVacuumTransmitterAbnormal = no1CondVacuumTransmitterAbnormal;  
 }  
  
   
 @Column(name="NO2\_COND\_VACUUM\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo2CondVacuumTransmitterAbnormal() {  
 return this.no2CondVacuumTransmitterAbnormal;  
 }  
   
 public void setNo2CondVacuumTransmitterAbnormal(String no2CondVacuumTransmitterAbnormal) {  
 this.no2CondVacuumTransmitterAbnormal = no2CondVacuumTransmitterAbnormal;  
 }  
  
   
 @Column(name="COND\_VACUUM\_TRANSMITTER\_SELECT\_0\_NO2\_1\_NO1", length=100)  
 public String getCondVacuumTransmitterSelect0No21No1() {  
 return this.condVacuumTransmitterSelect0No21No1;  
 }  
   
 public void setCondVacuumTransmitterSelect0No21No1(String condVacuumTransmitterSelect0No21No1) {  
 this.condVacuumTransmitterSelect0No21No1 = condVacuumTransmitterSelect0No21No1;  
 }  
  
   
 @Column(name="GT1\_GAS\_SELECT", length=100)  
 public String getGt1GasSelect() {  
 return this.gt1GasSelect;  
 }  
   
 public void setGt1GasSelect(String gt1GasSelect) {  
 this.gt1GasSelect = gt1GasSelect;  
 }  
  
   
 @Column(name="GT2\_GAS\_SELECT", length=100)  
 public String getGt2GasSelect() {  
 return this.gt2GasSelect;  
 }  
   
 public void setGt2GasSelect(String gt2GasSelect) {  
 this.gt2GasSelect = gt2GasSelect;  
 }  
  
   
 @Column(name="GT1\_OIL\_SELECT", length=100)  
 public String getGt1OilSelect() {  
 return this.gt1OilSelect;  
 }  
   
 public void setGt1OilSelect(String gt1OilSelect) {  
 this.gt1OilSelect = gt1OilSelect;  
 }  
  
   
 @Column(name="GT2\_OIL\_SELECT", length=100)  
 public String getGt2OilSelect() {  
 return this.gt2OilSelect;  
 }  
   
 public void setGt2OilSelect(String gt2OilSelect) {  
 this.gt2OilSelect = gt2OilSelect;  
 }  
  
   
 @Column(name="NO1\_GT\_FCB\_SELECT", length=100)  
 public String getNo1GtFcbSelect() {  
 return this.no1GtFcbSelect;  
 }  
   
 public void setNo1GtFcbSelect(String no1GtFcbSelect) {  
 this.no1GtFcbSelect = no1GtFcbSelect;  
 }  
  
   
 @Column(name="NO2\_GT\_FCB\_SELECT", length=100)  
 public String getNo2GtFcbSelect() {  
 return this.no2GtFcbSelect;  
 }  
   
 public void setNo2GtFcbSelect(String no2GtFcbSelect) {  
 this.no2GtFcbSelect = no2GtFcbSelect;  
 }  
  
   
 @Column(name="FCB\_COMMAND", length=100)  
 public String getFcbCommand() {  
 return this.fcbCommand;  
 }  
   
 public void setFcbCommand(String fcbCommand) {  
 this.fcbCommand = fcbCommand;  
 }  
  
   
 @Column(name="GT1\_GAS\_COMB", length=100)  
 public String getGt1GasComb() {  
 return this.gt1GasComb;  
 }  
   
 public void setGt1GasComb(String gt1GasComb) {  
 this.gt1GasComb = gt1GasComb;  
 }  
  
   
 @Column(name="GT2\_GAS\_COMB", length=100)  
 public String getGt2GasComb() {  
 return this.gt2GasComb;  
 }  
   
 public void setGt2GasComb(String gt2GasComb) {  
 this.gt2GasComb = gt2GasComb;  
 }  
  
   
 @Column(name="GT1\_OIL\_COMB", length=100)  
 public String getGt1OilComb() {  
 return this.gt1OilComb;  
 }  
   
 public void setGt1OilComb(String gt1OilComb) {  
 this.gt1OilComb = gt1OilComb;  
 }  
  
   
 @Column(name="GT2\_OIL\_COMB", length=100)  
 public String getGt2OilComb() {  
 return this.gt2OilComb;  
 }  
   
 public void setGt2OilComb(String gt2OilComb) {  
 this.gt2OilComb = gt2OilComb;  
 }  
  
   
 @Column(name="GT1\_ENCENDER\_COMB", length=100)  
 public String getGt1EncenderComb() {  
 return this.gt1EncenderComb;  
 }  
   
 public void setGt1EncenderComb(String gt1EncenderComb) {  
 this.gt1EncenderComb = gt1EncenderComb;  
 }  
  
   
 @Column(name="GT2\_ENCENDER\_COMB", length=100)  
 public String getGt2EncenderComb() {  
 return this.gt2EncenderComb;  
 }  
   
 public void setGt2EncenderComb(String gt2EncenderComb) {  
 this.gt2EncenderComb = gt2EncenderComb;  
 }  
  
   
 @Column(name="ACTUAL\_HIGHER\_HEATING\_VALUE\_WET", length=100)  
 public String getActualHigherHeatingValueWet() {  
 return this.actualHigherHeatingValueWet;  
 }  
   
 public void setActualHigherHeatingValueWet(String actualHigherHeatingValueWet) {  
 this.actualHigherHeatingValueWet = actualHigherHeatingValueWet;  
 }  
  
   
 @Column(name="ACTUAL\_LOWER\_HEATING\_VALUE\_WET", length=100)  
 public String getActualLowerHeatingValueWet() {  
 return this.actualLowerHeatingValueWet;  
 }  
   
 public void setActualLowerHeatingValueWet(String actualLowerHeatingValueWet) {  
 this.actualLowerHeatingValueWet = actualLowerHeatingValueWet;  
 }  
  
   
 @Column(name="ACTUAL\_HIGHER\_HEATING\_VALUE\_WET2", length=100)  
 public String getActualHigherHeatingValueWet2() {  
 return this.actualHigherHeatingValueWet2;  
 }  
   
 public void setActualHigherHeatingValueWet2(String actualHigherHeatingValueWet2) {  
 this.actualHigherHeatingValueWet2 = actualHigherHeatingValueWet2;  
 }  
  
   
 @Column(name="ACTUAL\_LOWER\_HEATING\_VALUE\_WET2", length=100)  
 public String getActualLowerHeatingValueWet2() {  
 return this.actualLowerHeatingValueWet2;  
 }  
   
 public void setActualLowerHeatingValueWet2(String actualLowerHeatingValueWet2) {  
 this.actualLowerHeatingValueWet2 = actualLowerHeatingValueWet2;  
 }  
  
   
 @Column(name="GAS\_VOLUME\_TOTAL", length=100)  
 public String getGasVolumeTotal() {  
 return this.gasVolumeTotal;  
 }  
   
 public void setGasVolumeTotal(String gasVolumeTotal) {  
 this.gasVolumeTotal = gasVolumeTotal;  
 }  
  
   
 @Column(name="GAS\_MASS\_TOTAL", length=100)  
 public String getGasMassTotal() {  
 return this.gasMassTotal;  
 }  
   
 public void setGasMassTotal(String gasMassTotal) {  
 this.gasMassTotal = gasMassTotal;  
 }  
  
   
 @Column(name="QG5\_5MIN\_HEAT\_INPUT\_WITH\_GAS", length=100)  
 public String getQg55minHeatInputWithGas() {  
 return this.qg55minHeatInputWithGas;  
 }  
   
 public void setQg55minHeatInputWithGas(String qg55minHeatInputWithGas) {  
 this.qg55minHeatInputWithGas = qg55minHeatInputWithGas;  
 }  
  
   
 @Column(name="QD5\_5MIN\_HEAT\_INPUT\_WITH\_OIL", length=100)  
 public String getQd55minHeatInputWithOil() {  
 return this.qd55minHeatInputWithOil;  
 }  
   
 public void setQd55minHeatInputWithOil(String qd55minHeatInputWithOil) {  
 this.qd55minHeatInputWithOil = qd55minHeatInputWithOil;  
 }  
  
   
 @Column(name="GT1\_OPERATION\_MODE", length=100)  
 public String getGt1OperationMode() {  
 return this.gt1OperationMode;  
 }  
   
 public void setGt1OperationMode(String gt1OperationMode) {  
 this.gt1OperationMode = gt1OperationMode;  
 }  
  
   
 @Column(name="GT2\_OPERATION\_MODE", length=100)  
 public String getGt2OperationMode() {  
 return this.gt2OperationMode;  
 }  
   
 public void setGt2OperationMode(String gt2OperationMode) {  
 this.gt2OperationMode = gt2OperationMode;  
 }  
  
   
 @Column(name="SPARE1", length=100)  
 public String getSpare1() {  
 return this.spare1;  
 }  
   
 public void setSpare1(String spare1) {  
 this.spare1 = spare1;  
 }  
  
   
 @Column(name="SPARE2", length=100)  
 public String getSpare2() {  
 return this.spare2;  
 }  
   
 public void setSpare2(String spare2) {  
 this.spare2 = spare2;  
 }  
  
   
 @Column(name="SPARE3", length=100)  
 public String getSpare3() {  
 return this.spare3;  
 }  
   
 public void setSpare3(String spare3) {  
 this.spare3 = spare3;  
 }  
  
   
 @Column(name="TIPO\_CARGA", length=100)  
 public String getTipoCarga() {  
 return this.tipoCarga;  
 }  
   
 public void setTipoCarga(String tipoCarga) {  
 this.tipoCarga = tipoCarga;  
 }  
  
   
 @Column(name="CARGA\_ID", precision=22, scale=0)  
 public BigDecimal getCargaId() {  
 return this.cargaId;  
 }  
   
 public void setCargaId(BigDecimal cargaId) {  
 this.cargaId = cargaId;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="HORA\_CARGA", length=11)  
 public Date getHoraCarga() {  
 return this.horaCarga;  
 }  
   
 public void setHoraCarga(Date horaCarga) {  
 this.horaCarga = horaCarga;  
 }  
  
   
 @Column(name="ESTATUS\_LIMPIO", precision=22, scale=0)  
 public BigDecimal getEstatusLimpio() {  
 return this.estatusLimpio;  
 }  
   
 public void setEstatusLimpio(BigDecimal estatusLimpio) {  
 this.estatusLimpio = estatusLimpio;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxCrudosLimpios.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.Date;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
import javax.persistence.UniqueConstraint;  
  
/\*\*  
 \* TxCrudosLimpios generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_CRUDOS\_LIMPIOS"  
 , uniqueConstraints = @UniqueConstraint(columnNames={"DIA", "FIRMA"})   
)  
public class TxCrudosLimpios implements java.io.Serializable {  
  
  
 private String firma;  
 private Date dia;  
 private String minuto;  
 private Short fuelSelect;  
 private BigDecimal hourlyLoadInUse;  
 private BigDecimal line1WattHourMas5min;  
 private BigDecimal line2WattHourMas5min;  
 private BigDecimal line1WattHourMenos5min;  
 private BigDecimal line2WattHourMenos5min;  
 private BigDecimal line1WattHourMas5minBackup;  
 private BigDecimal line2WattHourMas5minBackup;  
 private BigDecimal line1WattHourMenos5minBackup;  
 private BigDecimal line2WattHourMenos5minBackup;  
 private BigDecimal gt1ActivePower;  
 private BigDecimal gt2ActivePower;  
 private BigDecimal stActivePower;  
 private BigDecimal gt1ReactivePower;  
 private BigDecimal gt2ReactivePower;  
 private BigDecimal stReactivePower;  
 private BigDecimal line1WattInstant;  
 private BigDecimal line2WattInstant;  
 private BigDecimal line1VarInstant;  
 private BigDecimal line2VarInstant;  
 private BigDecimal plantNetWattHourCalculationFromGross;  
 private BigDecimal stgWattHour;  
 private BigDecimal stgVarHour;  
 private BigDecimal no1GtgWattHour;  
 private BigDecimal no1GtgVarHour;  
 private BigDecimal no1GtgUnitHouseLoad;  
 private BigDecimal no2GtgWattHour;  
 private BigDecimal no2GtgVarHour;  
 private BigDecimal no2GtgUnitHouseLoad;  
 private BigDecimal ambientTemperatureInUse;  
 private BigDecimal ambientTemperature1;  
 private BigDecimal ambientTemperature2;  
 private BigDecimal relativeHumidityInUse;  
 private BigDecimal relativeHumidity1;  
 private BigDecimal relativeHumidity2;  
 private BigDecimal ambientPressureInUse;  
 private BigDecimal ambientPressure1;  
 private BigDecimal ambientPressure2;  
 private BigDecimal condenserVacuumInUse;  
 private BigDecimal condenserVacuum1;  
 private BigDecimal condenserVacuum2;  
 private BigDecimal condenserVacuumMonitoring;  
 private BigDecimal fuelGasHigherHeatingValue;  
 private BigDecimal fuelGasLowerHeatingValue;  
 private BigDecimal fuelGasHigherHeatingValue2;  
 private BigDecimal fuelGasLowerHeatingValue2;  
 private BigDecimal fuelOilHigherHeatingValeu;  
 private BigDecimal fuelOilLowerHeatingValue;  
 private BigDecimal fuelOilHigherHeatingValeu2;  
 private BigDecimal fuelOilLowerHeatingValue2;  
 private BigDecimal fuelOilSpecificGravity;  
 private BigDecimal fuelGasFlowCurrentMasFlow;  
 private BigDecimal strm1CurrentVolumeFlowRate;  
 private BigDecimal fuelGasFlow5minAccumulatedValue;  
 private BigDecimal oilFlowGt1Gt2;  
 private BigDecimal gt1OilFlowCurrentValue;  
 private BigDecimal gt2OilFlowCurrentValue;  
 private BigDecimal oilFlowGt1Gt22;  
 private BigDecimal gt1OilFlow5minAccumulatedValue;  
 private BigDecimal gt2OilFlow5minAccumulatedValue;  
 private BigDecimal overallPowerFactor;  
 private BigDecimal netPowerOf5min5minAccumulatedWithGasFiring;  
 private BigDecimal netPowerOf5min5minAccumulatedWithOilFiring;  
 private BigDecimal loadPercentageAtGas;  
 private BigDecimal loadPercentageAtOil;  
 private BigDecimal ctungAtYearAverageDesignConditionGas;  
 private BigDecimal ctungAtYearAverageDesignConditionOil;  
 private BigDecimal correctedCtungGas;  
 private BigDecimal correctedCtungOil;  
 private BigDecimal fuelGasPressure;  
 private BigDecimal gasTemperature;  
 private BigDecimal loadDemand;  
 private BigDecimal eh512RollingAverage;  
 private BigDecimal ambientTemperature12RollingAverage;  
 private BigDecimal relativeHumidity12RollingAverage;  
 private BigDecimal ambientPressure12RollingAverage;  
 private BigDecimal condenserVacuum12RollingAverage;  
 private BigDecimal fuelGasFlow12RollingAverage;  
 private BigDecimal fuelGasFlow12RollingAverage2;  
 private BigDecimal fuelGasPressure12RollingAverage;  
 private BigDecimal overallPowerFactor12RollingAverage;  
 private BigDecimal fuelOilFlow12RollingAverage;  
 private BigDecimal fuelOilFlow12RollingAverage2;  
 private BigDecimal correctedCtung12RollingWeightedAverageGas;  
 private BigDecimal correctedCtung12RollingWeightedAverageOil;  
 private BigDecimal fuelGasHigherHeatingValue12RollingAverage;  
 private BigDecimal fuelGasLowerHeatingValue12RollingAverage;  
 private BigDecimal fuelGasHigherHeatingValue12RollingAverage2;  
 private BigDecimal fuelGasLowerHeatingValue12RollingAverage2;  
 private BigDecimal fuelOilHigherHeatingValue12RollingAverage;  
 private BigDecimal fuelOilLowerHeatingValue12RollingAverage;  
 private BigDecimal fuelOilHigherHeatingValue12RollingAverage2;  
 private BigDecimal fuelOilLowerHeatingValue12RollingAverage2;  
 private BigDecimal numberOfMinutesOfOperatingAtGasFiring;  
 private BigDecimal numberOfMinutesOfOperatingAtOilFiring;  
 private BigDecimal eh12RollingAverage;  
 private BigDecimal egh12RollingAverage;  
 private BigDecimal edh12RollingAverage;  
 private BigDecimal nhovgOperatingHourAtNoLoadBurningGas;  
 private BigDecimal nhovdOperatingHourAtNoLoadBurningOil;  
 private BigDecimal gt1OperatingHour;  
 private BigDecimal gt2OperatingHour;  
 private BigDecimal netOutputCorrectedToSummerDesignCfe;  
 private BigDecimal kcactualGas;  
 private BigDecimal kcactualOil;  
 private BigDecimal spare;  
 private BigDecimal gt152gOn;  
 private BigDecimal gt252gOn;  
 private BigDecimal EA1010400kvSSCbGt1Abn;  
 private BigDecimal EA1010400kvSSCbGt1Close;  
 private BigDecimal EA1010400kvSSCbGt1Open;  
 private BigDecimal EA1020400kvSSCbGt2Abn;  
 private BigDecimal EA1020400kvSSCbGt2Close;  
 private BigDecimal EA1020400kvSSCbGt2Open;  
 private BigDecimal EA1030400kvSSCbStClose;  
 private BigDecimal EA1030400kvSSCbStOpen;  
 private BigDecimal gt1Synchronize;  
 private BigDecimal gt2Synchronize;  
 private BigDecimal allTariffMeterAbnormal;  
 private BigDecimal gt1TransmitterWatAndVarAbnormal;  
 private BigDecimal gt2TransmitterWatAndVarAbnormal;  
 private BigDecimal stTransmitterWatAndVarAbnormal;  
 private BigDecimal no1TempTransmitterAbnormal;  
 private BigDecimal no2TempTransmitterAbnormal;  
 private BigDecimal tempTransmitterSelect0No21No1;  
 private BigDecimal no1HumidityTransmitterAbnormal;  
 private BigDecimal no2HumidityTransmitterAbnormal;  
 private BigDecimal humidityTransmitterSelect0No21No1;  
 private BigDecimal no1AmbientPressureTransmitterAbnormal;  
 private BigDecimal no2AmbientPressureTransmitterAbnormal;  
 private BigDecimal ambientPressureTransmitterSelect0No21No1;  
 private BigDecimal no1CondVacuumTransmitterAbnormal;  
 private BigDecimal no2CondVacuumTransmitterAbnormal;  
 private BigDecimal condVacuumTransmitterSelect0No21No1;  
 private BigDecimal gt1GasSelect;  
 private BigDecimal gt2GasSelect;  
 private BigDecimal gt1OilSelect;  
 private BigDecimal gt2OilSelect;  
 private BigDecimal no1GtFcbSelect;  
 private BigDecimal no2GtFcbSelect;  
 private BigDecimal fcbCommand;  
 private BigDecimal gt1GasComb;  
 private BigDecimal gt2GasComb;  
 private BigDecimal gt1OilComb;  
 private BigDecimal gt2OilComb;  
 private BigDecimal gt1EncenderComb;  
 private BigDecimal gt2EncenderComb;  
 private BigDecimal actualHigherHeatingValueWet;  
 private BigDecimal actualLowerHeatingValueWet;  
 private BigDecimal actualHigherHeatingValueWet2;  
 private BigDecimal actualLowerHeatingValueWet2;  
 private BigDecimal gasVolumeTotal;  
 private BigDecimal gasMassTotal;  
 private BigDecimal qg55minHeatInputWithGas;  
 private BigDecimal qd55minHeatInputWithOil;  
 private BigDecimal gt1OperationMode;  
 private BigDecimal gt2OperationMode;  
 private BigDecimal spare1;  
 private BigDecimal spare2;  
 private BigDecimal spare3;  
 private String tipoCarga;  
 private BigDecimal cargaId;  
 private Date horaCarga;  
 private BigDecimal estatusLimpio;  
 private BigDecimal crudosId;  
 private Boolean correcto;  
  
 public TxCrudosLimpios() {  
 }  
  
   
 public TxCrudosLimpios(Date dia, String minuto) {  
 this.dia = dia;  
 this.minuto = minuto;  
 }  
 public TxCrudosLimpios(Date dia, String minuto, Short fuelSelect, BigDecimal hourlyLoadInUse, BigDecimal line1WattHourMas5min, BigDecimal line2WattHourMas5min, BigDecimal line1WattHourMenos5min, BigDecimal line2WattHourMenos5min, BigDecimal line1WattHourMas5minBackup, BigDecimal line2WattHourMas5minBackup, BigDecimal line1WattHourMenos5minBackup, BigDecimal line2WattHourMenos5minBackup, BigDecimal gt1ActivePower, BigDecimal gt2ActivePower, BigDecimal stActivePower, BigDecimal gt1ReactivePower, BigDecimal gt2ReactivePower, BigDecimal stReactivePower, BigDecimal line1WattInstant, BigDecimal line2WattInstant, BigDecimal line1VarInstant, BigDecimal line2VarInstant, BigDecimal plantNetWattHourCalculationFromGross, BigDecimal stgWattHour, BigDecimal stgVarHour, BigDecimal no1GtgWattHour, BigDecimal no1GtgVarHour, BigDecimal no1GtgUnitHouseLoad, BigDecimal no2GtgWattHour, BigDecimal no2GtgVarHour, BigDecimal no2GtgUnitHouseLoad, BigDecimal ambientTemperatureInUse, BigDecimal ambientTemperature1, BigDecimal ambientTemperature2, BigDecimal relativeHumidityInUse, BigDecimal relativeHumidity1, BigDecimal relativeHumidity2, BigDecimal ambientPressureInUse, BigDecimal ambientPressure1, BigDecimal ambientPressure2, BigDecimal condenserVacuumInUse, BigDecimal condenserVacuum1, BigDecimal condenserVacuum2, BigDecimal condenserVacuumMonitoring, BigDecimal fuelGasHigherHeatingValue, BigDecimal fuelGasLowerHeatingValue, BigDecimal fuelGasHigherHeatingValue2, BigDecimal fuelGasLowerHeatingValue2, BigDecimal fuelOilHigherHeatingValeu, BigDecimal fuelOilLowerHeatingValue, BigDecimal fuelOilHigherHeatingValeu2, BigDecimal fuelOilLowerHeatingValue2, BigDecimal fuelOilSpecificGravity, BigDecimal fuelGasFlowCurrentMasFlow, BigDecimal strm1CurrentVolumeFlowRate, BigDecimal fuelGasFlow5minAccumulatedValue, BigDecimal oilFlowGt1Gt2, BigDecimal gt1OilFlowCurrentValue, BigDecimal gt2OilFlowCurrentValue, BigDecimal oilFlowGt1Gt22, BigDecimal gt1OilFlow5minAccumulatedValue, BigDecimal gt2OilFlow5minAccumulatedValue, BigDecimal overallPowerFactor, BigDecimal netPowerOf5min5minAccumulatedWithGasFiring, BigDecimal netPowerOf5min5minAccumulatedWithOilFiring, BigDecimal loadPercentageAtGas, BigDecimal loadPercentageAtOil, BigDecimal ctungAtYearAverageDesignConditionGas, BigDecimal ctungAtYearAverageDesignConditionOil, BigDecimal correctedCtungGas, BigDecimal correctedCtungOil, BigDecimal fuelGasPressure, BigDecimal gasTemperature, BigDecimal loadDemand, BigDecimal eh512RollingAverage, BigDecimal ambientTemperature12RollingAverage, BigDecimal relativeHumidity12RollingAverage, BigDecimal ambientPressure12RollingAverage, BigDecimal condenserVacuum12RollingAverage, BigDecimal fuelGasFlow12RollingAverage, BigDecimal fuelGasFlow12RollingAverage2, BigDecimal fuelGasPressure12RollingAverage, BigDecimal overallPowerFactor12RollingAverage, BigDecimal fuelOilFlow12RollingAverage, BigDecimal fuelOilFlow12RollingAverage2, BigDecimal correctedCtung12RollingWeightedAverageGas, BigDecimal correctedCtung12RollingWeightedAverageOil, BigDecimal fuelGasHigherHeatingValue12RollingAverage, BigDecimal fuelGasLowerHeatingValue12RollingAverage, BigDecimal fuelGasHigherHeatingValue12RollingAverage2, BigDecimal fuelGasLowerHeatingValue12RollingAverage2, BigDecimal fuelOilHigherHeatingValue12RollingAverage, BigDecimal fuelOilLowerHeatingValue12RollingAverage, BigDecimal fuelOilHigherHeatingValue12RollingAverage2, BigDecimal fuelOilLowerHeatingValue12RollingAverage2, BigDecimal numberOfMinutesOfOperatingAtGasFiring, BigDecimal numberOfMinutesOfOperatingAtOilFiring, BigDecimal eh12RollingAverage, BigDecimal egh12RollingAverage, BigDecimal edh12RollingAverage, BigDecimal nhovgOperatingHourAtNoLoadBurningGas, BigDecimal nhovdOperatingHourAtNoLoadBurningOil, BigDecimal gt1OperatingHour, BigDecimal gt2OperatingHour, BigDecimal netOutputCorrectedToSummerDesignCfe, BigDecimal kcactualGas, BigDecimal kcactualOil, BigDecimal spare, BigDecimal gt152gOn, BigDecimal gt252gOn, BigDecimal EA1010400kvSSCbGt1Abn, BigDecimal EA1010400kvSSCbGt1Close, BigDecimal EA1010400kvSSCbGt1Open, BigDecimal EA1020400kvSSCbGt2Abn, BigDecimal EA1020400kvSSCbGt2Close, BigDecimal EA1020400kvSSCbGt2Open, BigDecimal EA1030400kvSSCbStClose, BigDecimal EA1030400kvSSCbStOpen, BigDecimal gt1Synchronize, BigDecimal gt2Synchronize, BigDecimal allTariffMeterAbnormal, BigDecimal gt1TransmitterWatAndVarAbnormal, BigDecimal gt2TransmitterWatAndVarAbnormal, BigDecimal stTransmitterWatAndVarAbnormal, BigDecimal no1TempTransmitterAbnormal, BigDecimal no2TempTransmitterAbnormal, BigDecimal tempTransmitterSelect0No21No1, BigDecimal no1HumidityTransmitterAbnormal, BigDecimal no2HumidityTransmitterAbnormal, BigDecimal humidityTransmitterSelect0No21No1, BigDecimal no1AmbientPressureTransmitterAbnormal, BigDecimal no2AmbientPressureTransmitterAbnormal, BigDecimal ambientPressureTransmitterSelect0No21No1, BigDecimal no1CondVacuumTransmitterAbnormal, BigDecimal no2CondVacuumTransmitterAbnormal, BigDecimal condVacuumTransmitterSelect0No21No1, BigDecimal gt1GasSelect, BigDecimal gt2GasSelect, BigDecimal gt1OilSelect, BigDecimal gt2OilSelect, BigDecimal no1GtFcbSelect, BigDecimal no2GtFcbSelect, BigDecimal fcbCommand, BigDecimal gt1GasComb, BigDecimal gt2GasComb, BigDecimal gt1OilComb, BigDecimal gt2OilComb, BigDecimal gt1EncenderComb, BigDecimal gt2EncenderComb, BigDecimal actualHigherHeatingValueWet, BigDecimal actualLowerHeatingValueWet, BigDecimal actualHigherHeatingValueWet2, BigDecimal actualLowerHeatingValueWet2, BigDecimal gasVolumeTotal, BigDecimal gasMassTotal, BigDecimal qg55minHeatInputWithGas, BigDecimal qd55minHeatInputWithOil, BigDecimal gt1OperationMode, BigDecimal gt2OperationMode, BigDecimal spare1, BigDecimal spare2, BigDecimal spare3, String tipoCarga, BigDecimal cargaId, Date horaCarga, BigDecimal estatusLimpio, BigDecimal crudosId, Boolean correcto) {  
 this.dia = dia;  
 this.minuto = minuto;  
 this.fuelSelect = fuelSelect;  
 this.hourlyLoadInUse = hourlyLoadInUse;  
 this.line1WattHourMas5min = line1WattHourMas5min;  
 this.line2WattHourMas5min = line2WattHourMas5min;  
 this.line1WattHourMenos5min = line1WattHourMenos5min;  
 this.line2WattHourMenos5min = line2WattHourMenos5min;  
 this.line1WattHourMas5minBackup = line1WattHourMas5minBackup;  
 this.line2WattHourMas5minBackup = line2WattHourMas5minBackup;  
 this.line1WattHourMenos5minBackup = line1WattHourMenos5minBackup;  
 this.line2WattHourMenos5minBackup = line2WattHourMenos5minBackup;  
 this.gt1ActivePower = gt1ActivePower;  
 this.gt2ActivePower = gt2ActivePower;  
 this.stActivePower = stActivePower;  
 this.gt1ReactivePower = gt1ReactivePower;  
 this.gt2ReactivePower = gt2ReactivePower;  
 this.stReactivePower = stReactivePower;  
 this.line1WattInstant = line1WattInstant;  
 this.line2WattInstant = line2WattInstant;  
 this.line1VarInstant = line1VarInstant;  
 this.line2VarInstant = line2VarInstant;  
 this.plantNetWattHourCalculationFromGross = plantNetWattHourCalculationFromGross;  
 this.stgWattHour = stgWattHour;  
 this.stgVarHour = stgVarHour;  
 this.no1GtgWattHour = no1GtgWattHour;  
 this.no1GtgVarHour = no1GtgVarHour;  
 this.no1GtgUnitHouseLoad = no1GtgUnitHouseLoad;  
 this.no2GtgWattHour = no2GtgWattHour;  
 this.no2GtgVarHour = no2GtgVarHour;  
 this.no2GtgUnitHouseLoad = no2GtgUnitHouseLoad;  
 this.ambientTemperatureInUse = ambientTemperatureInUse;  
 this.ambientTemperature1 = ambientTemperature1;  
 this.ambientTemperature2 = ambientTemperature2;  
 this.relativeHumidityInUse = relativeHumidityInUse;  
 this.relativeHumidity1 = relativeHumidity1;  
 this.relativeHumidity2 = relativeHumidity2;  
 this.ambientPressureInUse = ambientPressureInUse;  
 this.ambientPressure1 = ambientPressure1;  
 this.ambientPressure2 = ambientPressure2;  
 this.condenserVacuumInUse = condenserVacuumInUse;  
 this.condenserVacuum1 = condenserVacuum1;  
 this.condenserVacuum2 = condenserVacuum2;  
 this.condenserVacuumMonitoring = condenserVacuumMonitoring;  
 this.fuelGasHigherHeatingValue = fuelGasHigherHeatingValue;  
 this.fuelGasLowerHeatingValue = fuelGasLowerHeatingValue;  
 this.fuelGasHigherHeatingValue2 = fuelGasHigherHeatingValue2;  
 this.fuelGasLowerHeatingValue2 = fuelGasLowerHeatingValue2;  
 this.fuelOilHigherHeatingValeu = fuelOilHigherHeatingValeu;  
 this.fuelOilLowerHeatingValue = fuelOilLowerHeatingValue;  
 this.fuelOilHigherHeatingValeu2 = fuelOilHigherHeatingValeu2;  
 this.fuelOilLowerHeatingValue2 = fuelOilLowerHeatingValue2;  
 this.fuelOilSpecificGravity = fuelOilSpecificGravity;  
 this.fuelGasFlowCurrentMasFlow = fuelGasFlowCurrentMasFlow;  
 this.strm1CurrentVolumeFlowRate = strm1CurrentVolumeFlowRate;  
 this.fuelGasFlow5minAccumulatedValue = fuelGasFlow5minAccumulatedValue;  
 this.oilFlowGt1Gt2 = oilFlowGt1Gt2;  
 this.gt1OilFlowCurrentValue = gt1OilFlowCurrentValue;  
 this.gt2OilFlowCurrentValue = gt2OilFlowCurrentValue;  
 this.oilFlowGt1Gt22 = oilFlowGt1Gt22;  
 this.gt1OilFlow5minAccumulatedValue = gt1OilFlow5minAccumulatedValue;  
 this.gt2OilFlow5minAccumulatedValue = gt2OilFlow5minAccumulatedValue;  
 this.overallPowerFactor = overallPowerFactor;  
 this.netPowerOf5min5minAccumulatedWithGasFiring = netPowerOf5min5minAccumulatedWithGasFiring;  
 this.netPowerOf5min5minAccumulatedWithOilFiring = netPowerOf5min5minAccumulatedWithOilFiring;  
 this.loadPercentageAtGas = loadPercentageAtGas;  
 this.loadPercentageAtOil = loadPercentageAtOil;  
 this.ctungAtYearAverageDesignConditionGas = ctungAtYearAverageDesignConditionGas;  
 this.ctungAtYearAverageDesignConditionOil = ctungAtYearAverageDesignConditionOil;  
 this.correctedCtungGas = correctedCtungGas;  
 this.correctedCtungOil = correctedCtungOil;  
 this.fuelGasPressure = fuelGasPressure;  
 this.gasTemperature = gasTemperature;  
 this.loadDemand = loadDemand;  
 this.eh512RollingAverage = eh512RollingAverage;  
 this.ambientTemperature12RollingAverage = ambientTemperature12RollingAverage;  
 this.relativeHumidity12RollingAverage = relativeHumidity12RollingAverage;  
 this.ambientPressure12RollingAverage = ambientPressure12RollingAverage;  
 this.condenserVacuum12RollingAverage = condenserVacuum12RollingAverage;  
 this.fuelGasFlow12RollingAverage = fuelGasFlow12RollingAverage;  
 this.fuelGasFlow12RollingAverage2 = fuelGasFlow12RollingAverage2;  
 this.fuelGasPressure12RollingAverage = fuelGasPressure12RollingAverage;  
 this.overallPowerFactor12RollingAverage = overallPowerFactor12RollingAverage;  
 this.fuelOilFlow12RollingAverage = fuelOilFlow12RollingAverage;  
 this.fuelOilFlow12RollingAverage2 = fuelOilFlow12RollingAverage2;  
 this.correctedCtung12RollingWeightedAverageGas = correctedCtung12RollingWeightedAverageGas;  
 this.correctedCtung12RollingWeightedAverageOil = correctedCtung12RollingWeightedAverageOil;  
 this.fuelGasHigherHeatingValue12RollingAverage = fuelGasHigherHeatingValue12RollingAverage;  
 this.fuelGasLowerHeatingValue12RollingAverage = fuelGasLowerHeatingValue12RollingAverage;  
 this.fuelGasHigherHeatingValue12RollingAverage2 = fuelGasHigherHeatingValue12RollingAverage2;  
 this.fuelGasLowerHeatingValue12RollingAverage2 = fuelGasLowerHeatingValue12RollingAverage2;  
 this.fuelOilHigherHeatingValue12RollingAverage = fuelOilHigherHeatingValue12RollingAverage;  
 this.fuelOilLowerHeatingValue12RollingAverage = fuelOilLowerHeatingValue12RollingAverage;  
 this.fuelOilHigherHeatingValue12RollingAverage2 = fuelOilHigherHeatingValue12RollingAverage2;  
 this.fuelOilLowerHeatingValue12RollingAverage2 = fuelOilLowerHeatingValue12RollingAverage2;  
 this.numberOfMinutesOfOperatingAtGasFiring = numberOfMinutesOfOperatingAtGasFiring;  
 this.numberOfMinutesOfOperatingAtOilFiring = numberOfMinutesOfOperatingAtOilFiring;  
 this.eh12RollingAverage = eh12RollingAverage;  
 this.egh12RollingAverage = egh12RollingAverage;  
 this.edh12RollingAverage = edh12RollingAverage;  
 this.nhovgOperatingHourAtNoLoadBurningGas = nhovgOperatingHourAtNoLoadBurningGas;  
 this.nhovdOperatingHourAtNoLoadBurningOil = nhovdOperatingHourAtNoLoadBurningOil;  
 this.gt1OperatingHour = gt1OperatingHour;  
 this.gt2OperatingHour = gt2OperatingHour;  
 this.netOutputCorrectedToSummerDesignCfe = netOutputCorrectedToSummerDesignCfe;  
 this.kcactualGas = kcactualGas;  
 this.kcactualOil = kcactualOil;  
 this.spare = spare;  
 this.gt152gOn = gt152gOn;  
 this.gt252gOn = gt252gOn;  
 this.EA1010400kvSSCbGt1Abn = EA1010400kvSSCbGt1Abn;  
 this.EA1010400kvSSCbGt1Close = EA1010400kvSSCbGt1Close;  
 this.EA1010400kvSSCbGt1Open = EA1010400kvSSCbGt1Open;  
 this.EA1020400kvSSCbGt2Abn = EA1020400kvSSCbGt2Abn;  
 this.EA1020400kvSSCbGt2Close = EA1020400kvSSCbGt2Close;  
 this.EA1020400kvSSCbGt2Open = EA1020400kvSSCbGt2Open;  
 this.EA1030400kvSSCbStClose = EA1030400kvSSCbStClose;  
 this.EA1030400kvSSCbStOpen = EA1030400kvSSCbStOpen;  
 this.gt1Synchronize = gt1Synchronize;  
 this.gt2Synchronize = gt2Synchronize;  
 this.allTariffMeterAbnormal = allTariffMeterAbnormal;  
 this.gt1TransmitterWatAndVarAbnormal = gt1TransmitterWatAndVarAbnormal;  
 this.gt2TransmitterWatAndVarAbnormal = gt2TransmitterWatAndVarAbnormal;  
 this.stTransmitterWatAndVarAbnormal = stTransmitterWatAndVarAbnormal;  
 this.no1TempTransmitterAbnormal = no1TempTransmitterAbnormal;  
 this.no2TempTransmitterAbnormal = no2TempTransmitterAbnormal;  
 this.tempTransmitterSelect0No21No1 = tempTransmitterSelect0No21No1;  
 this.no1HumidityTransmitterAbnormal = no1HumidityTransmitterAbnormal;  
 this.no2HumidityTransmitterAbnormal = no2HumidityTransmitterAbnormal;  
 this.humidityTransmitterSelect0No21No1 = humidityTransmitterSelect0No21No1;  
 this.no1AmbientPressureTransmitterAbnormal = no1AmbientPressureTransmitterAbnormal;  
 this.no2AmbientPressureTransmitterAbnormal = no2AmbientPressureTransmitterAbnormal;  
 this.ambientPressureTransmitterSelect0No21No1 = ambientPressureTransmitterSelect0No21No1;  
 this.no1CondVacuumTransmitterAbnormal = no1CondVacuumTransmitterAbnormal;  
 this.no2CondVacuumTransmitterAbnormal = no2CondVacuumTransmitterAbnormal;  
 this.condVacuumTransmitterSelect0No21No1 = condVacuumTransmitterSelect0No21No1;  
 this.gt1GasSelect = gt1GasSelect;  
 this.gt2GasSelect = gt2GasSelect;  
 this.gt1OilSelect = gt1OilSelect;  
 this.gt2OilSelect = gt2OilSelect;  
 this.no1GtFcbSelect = no1GtFcbSelect;  
 this.no2GtFcbSelect = no2GtFcbSelect;  
 this.fcbCommand = fcbCommand;  
 this.gt1GasComb = gt1GasComb;  
 this.gt2GasComb = gt2GasComb;  
 this.gt1OilComb = gt1OilComb;  
 this.gt2OilComb = gt2OilComb;  
 this.gt1EncenderComb = gt1EncenderComb;  
 this.gt2EncenderComb = gt2EncenderComb;  
 this.actualHigherHeatingValueWet = actualHigherHeatingValueWet;  
 this.actualLowerHeatingValueWet = actualLowerHeatingValueWet;  
 this.actualHigherHeatingValueWet2 = actualHigherHeatingValueWet2;  
 this.actualLowerHeatingValueWet2 = actualLowerHeatingValueWet2;  
 this.gasVolumeTotal = gasVolumeTotal;  
 this.gasMassTotal = gasMassTotal;  
 this.qg55minHeatInputWithGas = qg55minHeatInputWithGas;  
 this.qd55minHeatInputWithOil = qd55minHeatInputWithOil;  
 this.gt1OperationMode = gt1OperationMode;  
 this.gt2OperationMode = gt2OperationMode;  
 this.spare1 = spare1;  
 this.spare2 = spare2;  
 this.spare3 = spare3;  
 this.tipoCarga = tipoCarga;  
 this.cargaId = cargaId;  
 this.horaCarga = horaCarga;  
 this.estatusLimpio = estatusLimpio;  
 this.crudosId = crudosId;  
 this.correcto = correcto;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="FIRMA", unique=true, nullable=false, length=20)  
 public String getFirma() {  
 return this.firma;  
 }  
   
 public void setFirma(String firma) {  
 this.firma = firma;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="DIA", nullable=false, length=7)  
 public Date getDia() {  
 return this.dia;  
 }  
   
 public void setDia(Date dia) {  
 this.dia = dia;  
 }  
  
   
 @Column(name="MINUTO", nullable=false, length=26)  
 public String getMinuto() {  
 return this.minuto;  
 }  
   
 public void setMinuto(String minuto) {  
 this.minuto = minuto;  
 }  
  
   
 @Column(name="FUEL\_SELECT", precision=3, scale=0)  
 public Short getFuelSelect() {  
 return this.fuelSelect;  
 }  
   
 public void setFuelSelect(Short fuelSelect) {  
 this.fuelSelect = fuelSelect;  
 }  
  
   
 @Column(name="HOURLY\_LOAD\_IN\_USE", precision=20, scale=10)  
 public BigDecimal getHourlyLoadInUse() {  
 return this.hourlyLoadInUse;  
 }  
   
 public void setHourlyLoadInUse(BigDecimal hourlyLoadInUse) {  
 this.hourlyLoadInUse = hourlyLoadInUse;  
 }  
  
   
 @Column(name="LINE1\_WATT\_HOUR\_MAS\_5MIN", precision=20, scale=10)  
 public BigDecimal getLine1WattHourMas5min() {  
 return this.line1WattHourMas5min;  
 }  
   
 public void setLine1WattHourMas5min(BigDecimal line1WattHourMas5min) {  
 this.line1WattHourMas5min = line1WattHourMas5min;  
 }  
  
   
 @Column(name="LINE2\_WATT\_HOUR\_MAS\_5MIN", precision=20, scale=10)  
 public BigDecimal getLine2WattHourMas5min() {  
 return this.line2WattHourMas5min;  
 }  
   
 public void setLine2WattHourMas5min(BigDecimal line2WattHourMas5min) {  
 this.line2WattHourMas5min = line2WattHourMas5min;  
 }  
  
   
 @Column(name="LINE1\_WATT\_HOUR\_MENOS\_5MIN", precision=20, scale=10)  
 public BigDecimal getLine1WattHourMenos5min() {  
 return this.line1WattHourMenos5min;  
 }  
   
 public void setLine1WattHourMenos5min(BigDecimal line1WattHourMenos5min) {  
 this.line1WattHourMenos5min = line1WattHourMenos5min;  
 }  
  
   
 @Column(name="LINE2\_WATT\_HOUR\_MENOS\_5MIN", precision=20, scale=10)  
 public BigDecimal getLine2WattHourMenos5min() {  
 return this.line2WattHourMenos5min;  
 }  
   
 public void setLine2WattHourMenos5min(BigDecimal line2WattHourMenos5min) {  
 this.line2WattHourMenos5min = line2WattHourMenos5min;  
 }  
  
   
 @Column(name="LINE1\_WATT\_HOUR\_MAS\_5MIN\_BACKUP", precision=20, scale=10)  
 public BigDecimal getLine1WattHourMas5minBackup() {  
 return this.line1WattHourMas5minBackup;  
 }  
   
 public void setLine1WattHourMas5minBackup(BigDecimal line1WattHourMas5minBackup) {  
 this.line1WattHourMas5minBackup = line1WattHourMas5minBackup;  
 }  
  
   
 @Column(name="LINE2\_WATT\_HOUR\_MAS\_5MIN\_BACKUP", precision=20, scale=10)  
 public BigDecimal getLine2WattHourMas5minBackup() {  
 return this.line2WattHourMas5minBackup;  
 }  
   
 public void setLine2WattHourMas5minBackup(BigDecimal line2WattHourMas5minBackup) {  
 this.line2WattHourMas5minBackup = line2WattHourMas5minBackup;  
 }  
  
   
 @Column(name="LINE1\_WATT\_HOUR\_MENOS\_5MIN\_BACKUP", precision=20, scale=10)  
 public BigDecimal getLine1WattHourMenos5minBackup() {  
 return this.line1WattHourMenos5minBackup;  
 }  
   
 public void setLine1WattHourMenos5minBackup(BigDecimal line1WattHourMenos5minBackup) {  
 this.line1WattHourMenos5minBackup = line1WattHourMenos5minBackup;  
 }  
  
   
 @Column(name="LINE2\_WATT\_HOUR\_MENOS\_5MIN\_BACKUP", precision=20, scale=10)  
 public BigDecimal getLine2WattHourMenos5minBackup() {  
 return this.line2WattHourMenos5minBackup;  
 }  
   
 public void setLine2WattHourMenos5minBackup(BigDecimal line2WattHourMenos5minBackup) {  
 this.line2WattHourMenos5minBackup = line2WattHourMenos5minBackup;  
 }  
  
   
 @Column(name="GT1\_ACTIVE\_POWER", precision=20, scale=10)  
 public BigDecimal getGt1ActivePower() {  
 return this.gt1ActivePower;  
 }  
   
 public void setGt1ActivePower(BigDecimal gt1ActivePower) {  
 this.gt1ActivePower = gt1ActivePower;  
 }  
  
   
 @Column(name="GT2\_ACTIVE\_POWER", precision=20, scale=10)  
 public BigDecimal getGt2ActivePower() {  
 return this.gt2ActivePower;  
 }  
   
 public void setGt2ActivePower(BigDecimal gt2ActivePower) {  
 this.gt2ActivePower = gt2ActivePower;  
 }  
  
   
 @Column(name="ST\_ACTIVE\_POWER", precision=20, scale=10)  
 public BigDecimal getStActivePower() {  
 return this.stActivePower;  
 }  
   
 public void setStActivePower(BigDecimal stActivePower) {  
 this.stActivePower = stActivePower;  
 }  
  
   
 @Column(name="GT1\_REACTIVE\_POWER", precision=20, scale=10)  
 public BigDecimal getGt1ReactivePower() {  
 return this.gt1ReactivePower;  
 }  
   
 public void setGt1ReactivePower(BigDecimal gt1ReactivePower) {  
 this.gt1ReactivePower = gt1ReactivePower;  
 }  
  
   
 @Column(name="GT2\_REACTIVE\_POWER", precision=20, scale=10)  
 public BigDecimal getGt2ReactivePower() {  
 return this.gt2ReactivePower;  
 }  
   
 public void setGt2ReactivePower(BigDecimal gt2ReactivePower) {  
 this.gt2ReactivePower = gt2ReactivePower;  
 }  
  
   
 @Column(name="ST\_REACTIVE\_POWER", precision=20, scale=10)  
 public BigDecimal getStReactivePower() {  
 return this.stReactivePower;  
 }  
   
 public void setStReactivePower(BigDecimal stReactivePower) {  
 this.stReactivePower = stReactivePower;  
 }  
  
   
 @Column(name="LINE1\_WATT\_INSTANT", precision=20, scale=10)  
 public BigDecimal getLine1WattInstant() {  
 return this.line1WattInstant;  
 }  
   
 public void setLine1WattInstant(BigDecimal line1WattInstant) {  
 this.line1WattInstant = line1WattInstant;  
 }  
  
   
 @Column(name="LINE2\_WATT\_INSTANT", precision=20, scale=10)  
 public BigDecimal getLine2WattInstant() {  
 return this.line2WattInstant;  
 }  
   
 public void setLine2WattInstant(BigDecimal line2WattInstant) {  
 this.line2WattInstant = line2WattInstant;  
 }  
  
   
 @Column(name="LINE1\_VAR\_INSTANT", precision=20, scale=10)  
 public BigDecimal getLine1VarInstant() {  
 return this.line1VarInstant;  
 }  
   
 public void setLine1VarInstant(BigDecimal line1VarInstant) {  
 this.line1VarInstant = line1VarInstant;  
 }  
  
   
 @Column(name="LINE2\_VAR\_INSTANT", precision=20, scale=10)  
 public BigDecimal getLine2VarInstant() {  
 return this.line2VarInstant;  
 }  
   
 public void setLine2VarInstant(BigDecimal line2VarInstant) {  
 this.line2VarInstant = line2VarInstant;  
 }  
  
   
 @Column(name="PLANT\_NET\_WATT\_HOUR\_CALCULATION\_FROM\_GROSS", precision=20, scale=10)  
 public BigDecimal getPlantNetWattHourCalculationFromGross() {  
 return this.plantNetWattHourCalculationFromGross;  
 }  
   
 public void setPlantNetWattHourCalculationFromGross(BigDecimal plantNetWattHourCalculationFromGross) {  
 this.plantNetWattHourCalculationFromGross = plantNetWattHourCalculationFromGross;  
 }  
  
   
 @Column(name="STG\_WATT\_HOUR", precision=20, scale=10)  
 public BigDecimal getStgWattHour() {  
 return this.stgWattHour;  
 }  
   
 public void setStgWattHour(BigDecimal stgWattHour) {  
 this.stgWattHour = stgWattHour;  
 }  
  
   
 @Column(name="STG\_VAR\_HOUR", precision=20, scale=10)  
 public BigDecimal getStgVarHour() {  
 return this.stgVarHour;  
 }  
   
 public void setStgVarHour(BigDecimal stgVarHour) {  
 this.stgVarHour = stgVarHour;  
 }  
  
   
 @Column(name="NO1\_GTG\_WATT\_HOUR", precision=20, scale=10)  
 public BigDecimal getNo1GtgWattHour() {  
 return this.no1GtgWattHour;  
 }  
   
 public void setNo1GtgWattHour(BigDecimal no1GtgWattHour) {  
 this.no1GtgWattHour = no1GtgWattHour;  
 }  
  
   
 @Column(name="NO1\_GTG\_VAR\_HOUR", precision=20, scale=10)  
 public BigDecimal getNo1GtgVarHour() {  
 return this.no1GtgVarHour;  
 }  
   
 public void setNo1GtgVarHour(BigDecimal no1GtgVarHour) {  
 this.no1GtgVarHour = no1GtgVarHour;  
 }  
  
   
 @Column(name="NO1\_GTG\_UNIT\_HOUSE\_LOAD", precision=20, scale=10)  
 public BigDecimal getNo1GtgUnitHouseLoad() {  
 return this.no1GtgUnitHouseLoad;  
 }  
   
 public void setNo1GtgUnitHouseLoad(BigDecimal no1GtgUnitHouseLoad) {  
 this.no1GtgUnitHouseLoad = no1GtgUnitHouseLoad;  
 }  
  
   
 @Column(name="NO2\_GTG\_WATT\_HOUR", precision=20, scale=10)  
 public BigDecimal getNo2GtgWattHour() {  
 return this.no2GtgWattHour;  
 }  
   
 public void setNo2GtgWattHour(BigDecimal no2GtgWattHour) {  
 this.no2GtgWattHour = no2GtgWattHour;  
 }  
  
   
 @Column(name="NO2\_GTG\_VAR\_HOUR", precision=20, scale=10)  
 public BigDecimal getNo2GtgVarHour() {  
 return this.no2GtgVarHour;  
 }  
   
 public void setNo2GtgVarHour(BigDecimal no2GtgVarHour) {  
 this.no2GtgVarHour = no2GtgVarHour;  
 }  
  
   
 @Column(name="NO2\_GTG\_UNIT\_HOUSE\_LOAD", precision=20, scale=10)  
 public BigDecimal getNo2GtgUnitHouseLoad() {  
 return this.no2GtgUnitHouseLoad;  
 }  
   
 public void setNo2GtgUnitHouseLoad(BigDecimal no2GtgUnitHouseLoad) {  
 this.no2GtgUnitHouseLoad = no2GtgUnitHouseLoad;  
 }  
  
   
 @Column(name="AMBIENT\_TEMPERATURE\_IN\_USE", precision=20, scale=10)  
 public BigDecimal getAmbientTemperatureInUse() {  
 return this.ambientTemperatureInUse;  
 }  
   
 public void setAmbientTemperatureInUse(BigDecimal ambientTemperatureInUse) {  
 this.ambientTemperatureInUse = ambientTemperatureInUse;  
 }  
  
   
 @Column(name="AMBIENT\_TEMPERATURE1", precision=20, scale=10)  
 public BigDecimal getAmbientTemperature1() {  
 return this.ambientTemperature1;  
 }  
   
 public void setAmbientTemperature1(BigDecimal ambientTemperature1) {  
 this.ambientTemperature1 = ambientTemperature1;  
 }  
  
   
 @Column(name="AMBIENT\_TEMPERATURE2", precision=20, scale=10)  
 public BigDecimal getAmbientTemperature2() {  
 return this.ambientTemperature2;  
 }  
   
 public void setAmbientTemperature2(BigDecimal ambientTemperature2) {  
 this.ambientTemperature2 = ambientTemperature2;  
 }  
  
   
 @Column(name="RELATIVE\_HUMIDITY\_IN\_USE", precision=20, scale=10)  
 public BigDecimal getRelativeHumidityInUse() {  
 return this.relativeHumidityInUse;  
 }  
   
 public void setRelativeHumidityInUse(BigDecimal relativeHumidityInUse) {  
 this.relativeHumidityInUse = relativeHumidityInUse;  
 }  
  
   
 @Column(name="RELATIVE\_HUMIDITY1", precision=20, scale=10)  
 public BigDecimal getRelativeHumidity1() {  
 return this.relativeHumidity1;  
 }  
   
 public void setRelativeHumidity1(BigDecimal relativeHumidity1) {  
 this.relativeHumidity1 = relativeHumidity1;  
 }  
  
   
 @Column(name="RELATIVE\_HUMIDITY2", precision=20, scale=10)  
 public BigDecimal getRelativeHumidity2() {  
 return this.relativeHumidity2;  
 }  
   
 public void setRelativeHumidity2(BigDecimal relativeHumidity2) {  
 this.relativeHumidity2 = relativeHumidity2;  
 }  
  
   
 @Column(name="AMBIENT\_PRESSURE\_IN\_USE", precision=20, scale=10)  
 public BigDecimal getAmbientPressureInUse() {  
 return this.ambientPressureInUse;  
 }  
   
 public void setAmbientPressureInUse(BigDecimal ambientPressureInUse) {  
 this.ambientPressureInUse = ambientPressureInUse;  
 }  
  
   
 @Column(name="AMBIENT\_PRESSURE1", precision=20, scale=10)  
 public BigDecimal getAmbientPressure1() {  
 return this.ambientPressure1;  
 }  
   
 public void setAmbientPressure1(BigDecimal ambientPressure1) {  
 this.ambientPressure1 = ambientPressure1;  
 }  
  
   
 @Column(name="AMBIENT\_PRESSURE2", precision=20, scale=10)  
 public BigDecimal getAmbientPressure2() {  
 return this.ambientPressure2;  
 }  
   
 public void setAmbientPressure2(BigDecimal ambientPressure2) {  
 this.ambientPressure2 = ambientPressure2;  
 }  
  
   
 @Column(name="CONDENSER\_VACUUM\_IN\_USE", precision=20, scale=10)  
 public BigDecimal getCondenserVacuumInUse() {  
 return this.condenserVacuumInUse;  
 }  
   
 public void setCondenserVacuumInUse(BigDecimal condenserVacuumInUse) {  
 this.condenserVacuumInUse = condenserVacuumInUse;  
 }  
  
   
 @Column(name="CONDENSER\_VACUUM1", precision=20, scale=10)  
 public BigDecimal getCondenserVacuum1() {  
 return this.condenserVacuum1;  
 }  
   
 public void setCondenserVacuum1(BigDecimal condenserVacuum1) {  
 this.condenserVacuum1 = condenserVacuum1;  
 }  
  
   
 @Column(name="CONDENSER\_VACUUM2", precision=20, scale=10)  
 public BigDecimal getCondenserVacuum2() {  
 return this.condenserVacuum2;  
 }  
   
 public void setCondenserVacuum2(BigDecimal condenserVacuum2) {  
 this.condenserVacuum2 = condenserVacuum2;  
 }  
  
   
 @Column(name="CONDENSER\_VACUUM\_MONITORING", precision=20, scale=10)  
 public BigDecimal getCondenserVacuumMonitoring() {  
 return this.condenserVacuumMonitoring;  
 }  
   
 public void setCondenserVacuumMonitoring(BigDecimal condenserVacuumMonitoring) {  
 this.condenserVacuumMonitoring = condenserVacuumMonitoring;  
 }  
  
   
 @Column(name="FUEL\_GAS\_HIGHER\_HEATING\_VALUE", precision=20, scale=10)  
 public BigDecimal getFuelGasHigherHeatingValue() {  
 return this.fuelGasHigherHeatingValue;  
 }  
   
 public void setFuelGasHigherHeatingValue(BigDecimal fuelGasHigherHeatingValue) {  
 this.fuelGasHigherHeatingValue = fuelGasHigherHeatingValue;  
 }  
  
   
 @Column(name="FUEL\_GAS\_LOWER\_HEATING\_VALUE", precision=20, scale=10)  
 public BigDecimal getFuelGasLowerHeatingValue() {  
 return this.fuelGasLowerHeatingValue;  
 }  
   
 public void setFuelGasLowerHeatingValue(BigDecimal fuelGasLowerHeatingValue) {  
 this.fuelGasLowerHeatingValue = fuelGasLowerHeatingValue;  
 }  
  
   
 @Column(name="FUEL\_GAS\_HIGHER\_HEATING\_VALUE2", precision=20, scale=10)  
 public BigDecimal getFuelGasHigherHeatingValue2() {  
 return this.fuelGasHigherHeatingValue2;  
 }  
   
 public void setFuelGasHigherHeatingValue2(BigDecimal fuelGasHigherHeatingValue2) {  
 this.fuelGasHigherHeatingValue2 = fuelGasHigherHeatingValue2;  
 }  
  
   
 @Column(name="FUEL\_GAS\_LOWER\_HEATING\_VALUE2", precision=20, scale=10)  
 public BigDecimal getFuelGasLowerHeatingValue2() {  
 return this.fuelGasLowerHeatingValue2;  
 }  
   
 public void setFuelGasLowerHeatingValue2(BigDecimal fuelGasLowerHeatingValue2) {  
 this.fuelGasLowerHeatingValue2 = fuelGasLowerHeatingValue2;  
 }  
  
   
 @Column(name="FUEL\_OIL\_HIGHER\_HEATING\_VALEU", precision=20, scale=10)  
 public BigDecimal getFuelOilHigherHeatingValeu() {  
 return this.fuelOilHigherHeatingValeu;  
 }  
   
 public void setFuelOilHigherHeatingValeu(BigDecimal fuelOilHigherHeatingValeu) {  
 this.fuelOilHigherHeatingValeu = fuelOilHigherHeatingValeu;  
 }  
  
   
 @Column(name="FUEL\_OIL\_LOWER\_HEATING\_VALUE", precision=20, scale=10)  
 public BigDecimal getFuelOilLowerHeatingValue() {  
 return this.fuelOilLowerHeatingValue;  
 }  
   
 public void setFuelOilLowerHeatingValue(BigDecimal fuelOilLowerHeatingValue) {  
 this.fuelOilLowerHeatingValue = fuelOilLowerHeatingValue;  
 }  
  
   
 @Column(name="FUEL\_OIL\_HIGHER\_HEATING\_VALEU2", precision=20, scale=10)  
 public BigDecimal getFuelOilHigherHeatingValeu2() {  
 return this.fuelOilHigherHeatingValeu2;  
 }  
   
 public void setFuelOilHigherHeatingValeu2(BigDecimal fuelOilHigherHeatingValeu2) {  
 this.fuelOilHigherHeatingValeu2 = fuelOilHigherHeatingValeu2;  
 }  
  
   
 @Column(name="FUEL\_OIL\_LOWER\_HEATING\_VALUE2", precision=20, scale=10)  
 public BigDecimal getFuelOilLowerHeatingValue2() {  
 return this.fuelOilLowerHeatingValue2;  
 }  
   
 public void setFuelOilLowerHeatingValue2(BigDecimal fuelOilLowerHeatingValue2) {  
 this.fuelOilLowerHeatingValue2 = fuelOilLowerHeatingValue2;  
 }  
  
   
 @Column(name="FUEL\_OIL\_SPECIFIC\_GRAVITY", precision=20, scale=10)  
 public BigDecimal getFuelOilSpecificGravity() {  
 return this.fuelOilSpecificGravity;  
 }  
   
 public void setFuelOilSpecificGravity(BigDecimal fuelOilSpecificGravity) {  
 this.fuelOilSpecificGravity = fuelOilSpecificGravity;  
 }  
  
   
 @Column(name="FUEL\_GAS\_FLOW\_CURRENT\_MAS\_FLOW", precision=20, scale=10)  
 public BigDecimal getFuelGasFlowCurrentMasFlow() {  
 return this.fuelGasFlowCurrentMasFlow;  
 }  
   
 public void setFuelGasFlowCurrentMasFlow(BigDecimal fuelGasFlowCurrentMasFlow) {  
 this.fuelGasFlowCurrentMasFlow = fuelGasFlowCurrentMasFlow;  
 }  
  
   
 @Column(name="STRM1\_CURRENT\_VOLUME\_FLOW\_RATE", precision=20, scale=10)  
 public BigDecimal getStrm1CurrentVolumeFlowRate() {  
 return this.strm1CurrentVolumeFlowRate;  
 }  
   
 public void setStrm1CurrentVolumeFlowRate(BigDecimal strm1CurrentVolumeFlowRate) {  
 this.strm1CurrentVolumeFlowRate = strm1CurrentVolumeFlowRate;  
 }  
  
   
 @Column(name="FUEL\_GAS\_FLOW\_5MIN\_ACCUMULATED\_VALUE", precision=20, scale=10)  
 public BigDecimal getFuelGasFlow5minAccumulatedValue() {  
 return this.fuelGasFlow5minAccumulatedValue;  
 }  
   
 public void setFuelGasFlow5minAccumulatedValue(BigDecimal fuelGasFlow5minAccumulatedValue) {  
 this.fuelGasFlow5minAccumulatedValue = fuelGasFlow5minAccumulatedValue;  
 }  
  
   
 @Column(name="OIL\_FLOW\_GT1\_GT2", precision=20, scale=10)  
 public BigDecimal getOilFlowGt1Gt2() {  
 return this.oilFlowGt1Gt2;  
 }  
   
 public void setOilFlowGt1Gt2(BigDecimal oilFlowGt1Gt2) {  
 this.oilFlowGt1Gt2 = oilFlowGt1Gt2;  
 }  
  
   
 @Column(name="GT1\_OIL\_FLOW\_CURRENT\_VALUE", precision=20, scale=10)  
 public BigDecimal getGt1OilFlowCurrentValue() {  
 return this.gt1OilFlowCurrentValue;  
 }  
   
 public void setGt1OilFlowCurrentValue(BigDecimal gt1OilFlowCurrentValue) {  
 this.gt1OilFlowCurrentValue = gt1OilFlowCurrentValue;  
 }  
  
   
 @Column(name="GT2\_OIL\_FLOW\_CURRENT\_VALUE", precision=20, scale=10)  
 public BigDecimal getGt2OilFlowCurrentValue() {  
 return this.gt2OilFlowCurrentValue;  
 }  
   
 public void setGt2OilFlowCurrentValue(BigDecimal gt2OilFlowCurrentValue) {  
 this.gt2OilFlowCurrentValue = gt2OilFlowCurrentValue;  
 }  
  
   
 @Column(name="OIL\_FLOW\_GT1\_GT22", precision=20, scale=10)  
 public BigDecimal getOilFlowGt1Gt22() {  
 return this.oilFlowGt1Gt22;  
 }  
   
 public void setOilFlowGt1Gt22(BigDecimal oilFlowGt1Gt22) {  
 this.oilFlowGt1Gt22 = oilFlowGt1Gt22;  
 }  
  
   
 @Column(name="GT1\_OIL\_FLOW\_5MIN\_ACCUMULATED\_VALUE", precision=20, scale=10)  
 public BigDecimal getGt1OilFlow5minAccumulatedValue() {  
 return this.gt1OilFlow5minAccumulatedValue;  
 }  
   
 public void setGt1OilFlow5minAccumulatedValue(BigDecimal gt1OilFlow5minAccumulatedValue) {  
 this.gt1OilFlow5minAccumulatedValue = gt1OilFlow5minAccumulatedValue;  
 }  
  
   
 @Column(name="GT2\_OIL\_FLOW\_5MIN\_ACCUMULATED\_VALUE", precision=20, scale=10)  
 public BigDecimal getGt2OilFlow5minAccumulatedValue() {  
 return this.gt2OilFlow5minAccumulatedValue;  
 }  
   
 public void setGt2OilFlow5minAccumulatedValue(BigDecimal gt2OilFlow5minAccumulatedValue) {  
 this.gt2OilFlow5minAccumulatedValue = gt2OilFlow5minAccumulatedValue;  
 }  
  
   
 @Column(name="OVERALL\_POWER\_FACTOR", precision=20, scale=10)  
 public BigDecimal getOverallPowerFactor() {  
 return this.overallPowerFactor;  
 }  
   
 public void setOverallPowerFactor(BigDecimal overallPowerFactor) {  
 this.overallPowerFactor = overallPowerFactor;  
 }  
  
   
 @Column(name="NET\_POWER\_OF\_5MIN\_5MIN\_ACCUMULATED\_WITH\_GAS\_FIRING", precision=20, scale=10)  
 public BigDecimal getNetPowerOf5min5minAccumulatedWithGasFiring() {  
 return this.netPowerOf5min5minAccumulatedWithGasFiring;  
 }  
   
 public void setNetPowerOf5min5minAccumulatedWithGasFiring(BigDecimal netPowerOf5min5minAccumulatedWithGasFiring) {  
 this.netPowerOf5min5minAccumulatedWithGasFiring = netPowerOf5min5minAccumulatedWithGasFiring;  
 }  
  
   
 @Column(name="NET\_POWER\_OF\_5MIN\_5MIN\_ACCUMULATED\_WITH\_OIL\_FIRING", precision=20, scale=10)  
 public BigDecimal getNetPowerOf5min5minAccumulatedWithOilFiring() {  
 return this.netPowerOf5min5minAccumulatedWithOilFiring;  
 }  
   
 public void setNetPowerOf5min5minAccumulatedWithOilFiring(BigDecimal netPowerOf5min5minAccumulatedWithOilFiring) {  
 this.netPowerOf5min5minAccumulatedWithOilFiring = netPowerOf5min5minAccumulatedWithOilFiring;  
 }  
  
   
 @Column(name="LOAD\_PERCENTAGE\_AT\_GAS", precision=20, scale=10)  
 public BigDecimal getLoadPercentageAtGas() {  
 return this.loadPercentageAtGas;  
 }  
   
 public void setLoadPercentageAtGas(BigDecimal loadPercentageAtGas) {  
 this.loadPercentageAtGas = loadPercentageAtGas;  
 }  
  
   
 @Column(name="LOAD\_PERCENTAGE\_AT\_OIL", precision=20, scale=10)  
 public BigDecimal getLoadPercentageAtOil() {  
 return this.loadPercentageAtOil;  
 }  
   
 public void setLoadPercentageAtOil(BigDecimal loadPercentageAtOil) {  
 this.loadPercentageAtOil = loadPercentageAtOil;  
 }  
  
   
 @Column(name="CTUNG\_AT\_YEAR\_AVERAGE\_DESIGN\_CONDITION\_GAS", precision=20, scale=10)  
 public BigDecimal getCtungAtYearAverageDesignConditionGas() {  
 return this.ctungAtYearAverageDesignConditionGas;  
 }  
   
 public void setCtungAtYearAverageDesignConditionGas(BigDecimal ctungAtYearAverageDesignConditionGas) {  
 this.ctungAtYearAverageDesignConditionGas = ctungAtYearAverageDesignConditionGas;  
 }  
  
   
 @Column(name="CTUNG\_AT\_YEAR\_AVERAGE\_DESIGN\_CONDITION\_OIL", precision=20, scale=10)  
 public BigDecimal getCtungAtYearAverageDesignConditionOil() {  
 return this.ctungAtYearAverageDesignConditionOil;  
 }  
   
 public void setCtungAtYearAverageDesignConditionOil(BigDecimal ctungAtYearAverageDesignConditionOil) {  
 this.ctungAtYearAverageDesignConditionOil = ctungAtYearAverageDesignConditionOil;  
 }  
  
   
 @Column(name="CORRECTED\_CTUNG\_GAS", precision=20, scale=10)  
 public BigDecimal getCorrectedCtungGas() {  
 return this.correctedCtungGas;  
 }  
   
 public void setCorrectedCtungGas(BigDecimal correctedCtungGas) {  
 this.correctedCtungGas = correctedCtungGas;  
 }  
  
   
 @Column(name="CORRECTED\_CTUNG\_OIL", precision=20, scale=10)  
 public BigDecimal getCorrectedCtungOil() {  
 return this.correctedCtungOil;  
 }  
   
 public void setCorrectedCtungOil(BigDecimal correctedCtungOil) {  
 this.correctedCtungOil = correctedCtungOil;  
 }  
  
   
 @Column(name="FUEL\_GAS\_PRESSURE", precision=20, scale=10)  
 public BigDecimal getFuelGasPressure() {  
 return this.fuelGasPressure;  
 }  
   
 public void setFuelGasPressure(BigDecimal fuelGasPressure) {  
 this.fuelGasPressure = fuelGasPressure;  
 }  
  
   
 @Column(name="GAS\_TEMPERATURE", precision=20, scale=10)  
 public BigDecimal getGasTemperature() {  
 return this.gasTemperature;  
 }  
   
 public void setGasTemperature(BigDecimal gasTemperature) {  
 this.gasTemperature = gasTemperature;  
 }  
  
   
 @Column(name="LOAD\_DEMAND", precision=20, scale=10)  
 public BigDecimal getLoadDemand() {  
 return this.loadDemand;  
 }  
   
 public void setLoadDemand(BigDecimal loadDemand) {  
 this.loadDemand = loadDemand;  
 }  
  
   
 @Column(name="EH5\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getEh512RollingAverage() {  
 return this.eh512RollingAverage;  
 }  
   
 public void setEh512RollingAverage(BigDecimal eh512RollingAverage) {  
 this.eh512RollingAverage = eh512RollingAverage;  
 }  
  
   
 @Column(name="AMBIENT\_TEMPERATURE\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getAmbientTemperature12RollingAverage() {  
 return this.ambientTemperature12RollingAverage;  
 }  
   
 public void setAmbientTemperature12RollingAverage(BigDecimal ambientTemperature12RollingAverage) {  
 this.ambientTemperature12RollingAverage = ambientTemperature12RollingAverage;  
 }  
  
   
 @Column(name="RELATIVE\_HUMIDITY\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getRelativeHumidity12RollingAverage() {  
 return this.relativeHumidity12RollingAverage;  
 }  
   
 public void setRelativeHumidity12RollingAverage(BigDecimal relativeHumidity12RollingAverage) {  
 this.relativeHumidity12RollingAverage = relativeHumidity12RollingAverage;  
 }  
  
   
 @Column(name="AMBIENT\_PRESSURE\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getAmbientPressure12RollingAverage() {  
 return this.ambientPressure12RollingAverage;  
 }  
   
 public void setAmbientPressure12RollingAverage(BigDecimal ambientPressure12RollingAverage) {  
 this.ambientPressure12RollingAverage = ambientPressure12RollingAverage;  
 }  
  
   
 @Column(name="CONDENSER\_VACUUM\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getCondenserVacuum12RollingAverage() {  
 return this.condenserVacuum12RollingAverage;  
 }  
   
 public void setCondenserVacuum12RollingAverage(BigDecimal condenserVacuum12RollingAverage) {  
 this.condenserVacuum12RollingAverage = condenserVacuum12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_GAS\_FLOW\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getFuelGasFlow12RollingAverage() {  
 return this.fuelGasFlow12RollingAverage;  
 }  
   
 public void setFuelGasFlow12RollingAverage(BigDecimal fuelGasFlow12RollingAverage) {  
 this.fuelGasFlow12RollingAverage = fuelGasFlow12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_GAS\_FLOW\_12\_ROLLING\_AVERAGE2", precision=20, scale=10)  
 public BigDecimal getFuelGasFlow12RollingAverage2() {  
 return this.fuelGasFlow12RollingAverage2;  
 }  
   
 public void setFuelGasFlow12RollingAverage2(BigDecimal fuelGasFlow12RollingAverage2) {  
 this.fuelGasFlow12RollingAverage2 = fuelGasFlow12RollingAverage2;  
 }  
  
   
 @Column(name="FUEL\_GAS\_PRESSURE\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getFuelGasPressure12RollingAverage() {  
 return this.fuelGasPressure12RollingAverage;  
 }  
   
 public void setFuelGasPressure12RollingAverage(BigDecimal fuelGasPressure12RollingAverage) {  
 this.fuelGasPressure12RollingAverage = fuelGasPressure12RollingAverage;  
 }  
  
   
 @Column(name="OVERALL\_POWER\_FACTOR\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getOverallPowerFactor12RollingAverage() {  
 return this.overallPowerFactor12RollingAverage;  
 }  
   
 public void setOverallPowerFactor12RollingAverage(BigDecimal overallPowerFactor12RollingAverage) {  
 this.overallPowerFactor12RollingAverage = overallPowerFactor12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_OIL\_FLOW\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getFuelOilFlow12RollingAverage() {  
 return this.fuelOilFlow12RollingAverage;  
 }  
   
 public void setFuelOilFlow12RollingAverage(BigDecimal fuelOilFlow12RollingAverage) {  
 this.fuelOilFlow12RollingAverage = fuelOilFlow12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_OIL\_FLOW\_12\_ROLLING\_AVERAGE2", precision=20, scale=10)  
 public BigDecimal getFuelOilFlow12RollingAverage2() {  
 return this.fuelOilFlow12RollingAverage2;  
 }  
   
 public void setFuelOilFlow12RollingAverage2(BigDecimal fuelOilFlow12RollingAverage2) {  
 this.fuelOilFlow12RollingAverage2 = fuelOilFlow12RollingAverage2;  
 }  
  
   
 @Column(name="CORRECTED\_CTUNG\_12\_ROLLING\_WEIGHTED\_AVERAGE\_GAS", precision=20, scale=10)  
 public BigDecimal getCorrectedCtung12RollingWeightedAverageGas() {  
 return this.correctedCtung12RollingWeightedAverageGas;  
 }  
   
 public void setCorrectedCtung12RollingWeightedAverageGas(BigDecimal correctedCtung12RollingWeightedAverageGas) {  
 this.correctedCtung12RollingWeightedAverageGas = correctedCtung12RollingWeightedAverageGas;  
 }  
  
   
 @Column(name="CORRECTED\_CTUNG\_12\_ROLLING\_WEIGHTED\_AVERAGE\_OIL", precision=20, scale=10)  
 public BigDecimal getCorrectedCtung12RollingWeightedAverageOil() {  
 return this.correctedCtung12RollingWeightedAverageOil;  
 }  
   
 public void setCorrectedCtung12RollingWeightedAverageOil(BigDecimal correctedCtung12RollingWeightedAverageOil) {  
 this.correctedCtung12RollingWeightedAverageOil = correctedCtung12RollingWeightedAverageOil;  
 }  
  
   
 @Column(name="FUEL\_GAS\_HIGHER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getFuelGasHigherHeatingValue12RollingAverage() {  
 return this.fuelGasHigherHeatingValue12RollingAverage;  
 }  
   
 public void setFuelGasHigherHeatingValue12RollingAverage(BigDecimal fuelGasHigherHeatingValue12RollingAverage) {  
 this.fuelGasHigherHeatingValue12RollingAverage = fuelGasHigherHeatingValue12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_GAS\_LOWER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getFuelGasLowerHeatingValue12RollingAverage() {  
 return this.fuelGasLowerHeatingValue12RollingAverage;  
 }  
   
 public void setFuelGasLowerHeatingValue12RollingAverage(BigDecimal fuelGasLowerHeatingValue12RollingAverage) {  
 this.fuelGasLowerHeatingValue12RollingAverage = fuelGasLowerHeatingValue12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_GAS\_HIGHER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE2", precision=20, scale=10)  
 public BigDecimal getFuelGasHigherHeatingValue12RollingAverage2() {  
 return this.fuelGasHigherHeatingValue12RollingAverage2;  
 }  
   
 public void setFuelGasHigherHeatingValue12RollingAverage2(BigDecimal fuelGasHigherHeatingValue12RollingAverage2) {  
 this.fuelGasHigherHeatingValue12RollingAverage2 = fuelGasHigherHeatingValue12RollingAverage2;  
 }  
  
   
 @Column(name="FUEL\_GAS\_LOWER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE2", precision=20, scale=10)  
 public BigDecimal getFuelGasLowerHeatingValue12RollingAverage2() {  
 return this.fuelGasLowerHeatingValue12RollingAverage2;  
 }  
   
 public void setFuelGasLowerHeatingValue12RollingAverage2(BigDecimal fuelGasLowerHeatingValue12RollingAverage2) {  
 this.fuelGasLowerHeatingValue12RollingAverage2 = fuelGasLowerHeatingValue12RollingAverage2;  
 }  
  
   
 @Column(name="FUEL\_OIL\_HIGHER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getFuelOilHigherHeatingValue12RollingAverage() {  
 return this.fuelOilHigherHeatingValue12RollingAverage;  
 }  
   
 public void setFuelOilHigherHeatingValue12RollingAverage(BigDecimal fuelOilHigherHeatingValue12RollingAverage) {  
 this.fuelOilHigherHeatingValue12RollingAverage = fuelOilHigherHeatingValue12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_OIL\_LOWER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getFuelOilLowerHeatingValue12RollingAverage() {  
 return this.fuelOilLowerHeatingValue12RollingAverage;  
 }  
   
 public void setFuelOilLowerHeatingValue12RollingAverage(BigDecimal fuelOilLowerHeatingValue12RollingAverage) {  
 this.fuelOilLowerHeatingValue12RollingAverage = fuelOilLowerHeatingValue12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_OIL\_HIGHER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE2", precision=20, scale=10)  
 public BigDecimal getFuelOilHigherHeatingValue12RollingAverage2() {  
 return this.fuelOilHigherHeatingValue12RollingAverage2;  
 }  
   
 public void setFuelOilHigherHeatingValue12RollingAverage2(BigDecimal fuelOilHigherHeatingValue12RollingAverage2) {  
 this.fuelOilHigherHeatingValue12RollingAverage2 = fuelOilHigherHeatingValue12RollingAverage2;  
 }  
  
   
 @Column(name="FUEL\_OIL\_LOWER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE2", precision=20, scale=10)  
 public BigDecimal getFuelOilLowerHeatingValue12RollingAverage2() {  
 return this.fuelOilLowerHeatingValue12RollingAverage2;  
 }  
   
 public void setFuelOilLowerHeatingValue12RollingAverage2(BigDecimal fuelOilLowerHeatingValue12RollingAverage2) {  
 this.fuelOilLowerHeatingValue12RollingAverage2 = fuelOilLowerHeatingValue12RollingAverage2;  
 }  
  
   
 @Column(name="NUMBER\_OF\_MINUTES\_OF\_OPERATING\_AT\_GAS\_FIRING", precision=20, scale=10)  
 public BigDecimal getNumberOfMinutesOfOperatingAtGasFiring() {  
 return this.numberOfMinutesOfOperatingAtGasFiring;  
 }  
   
 public void setNumberOfMinutesOfOperatingAtGasFiring(BigDecimal numberOfMinutesOfOperatingAtGasFiring) {  
 this.numberOfMinutesOfOperatingAtGasFiring = numberOfMinutesOfOperatingAtGasFiring;  
 }  
  
   
 @Column(name="NUMBER\_OF\_MINUTES\_OF\_OPERATING\_AT\_OIL\_FIRING", precision=20, scale=10)  
 public BigDecimal getNumberOfMinutesOfOperatingAtOilFiring() {  
 return this.numberOfMinutesOfOperatingAtOilFiring;  
 }  
   
 public void setNumberOfMinutesOfOperatingAtOilFiring(BigDecimal numberOfMinutesOfOperatingAtOilFiring) {  
 this.numberOfMinutesOfOperatingAtOilFiring = numberOfMinutesOfOperatingAtOilFiring;  
 }  
  
   
 @Column(name="EH\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getEh12RollingAverage() {  
 return this.eh12RollingAverage;  
 }  
   
 public void setEh12RollingAverage(BigDecimal eh12RollingAverage) {  
 this.eh12RollingAverage = eh12RollingAverage;  
 }  
  
   
 @Column(name="EGH\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getEgh12RollingAverage() {  
 return this.egh12RollingAverage;  
 }  
   
 public void setEgh12RollingAverage(BigDecimal egh12RollingAverage) {  
 this.egh12RollingAverage = egh12RollingAverage;  
 }  
  
   
 @Column(name="EDH\_12\_ROLLING\_AVERAGE", precision=20, scale=10)  
 public BigDecimal getEdh12RollingAverage() {  
 return this.edh12RollingAverage;  
 }  
   
 public void setEdh12RollingAverage(BigDecimal edh12RollingAverage) {  
 this.edh12RollingAverage = edh12RollingAverage;  
 }  
  
   
 @Column(name="NHOVG\_OPERATING\_HOUR\_AT\_NO\_LOAD\_BURNING\_GAS", precision=20, scale=10)  
 public BigDecimal getNhovgOperatingHourAtNoLoadBurningGas() {  
 return this.nhovgOperatingHourAtNoLoadBurningGas;  
 }  
   
 public void setNhovgOperatingHourAtNoLoadBurningGas(BigDecimal nhovgOperatingHourAtNoLoadBurningGas) {  
 this.nhovgOperatingHourAtNoLoadBurningGas = nhovgOperatingHourAtNoLoadBurningGas;  
 }  
  
   
 @Column(name="NHOVD\_OPERATING\_HOUR\_AT\_NO\_LOAD\_BURNING\_OIL", precision=20, scale=10)  
 public BigDecimal getNhovdOperatingHourAtNoLoadBurningOil() {  
 return this.nhovdOperatingHourAtNoLoadBurningOil;  
 }  
   
 public void setNhovdOperatingHourAtNoLoadBurningOil(BigDecimal nhovdOperatingHourAtNoLoadBurningOil) {  
 this.nhovdOperatingHourAtNoLoadBurningOil = nhovdOperatingHourAtNoLoadBurningOil;  
 }  
  
   
 @Column(name="GT1\_OPERATING\_HOUR", precision=20, scale=10)  
 public BigDecimal getGt1OperatingHour() {  
 return this.gt1OperatingHour;  
 }  
   
 public void setGt1OperatingHour(BigDecimal gt1OperatingHour) {  
 this.gt1OperatingHour = gt1OperatingHour;  
 }  
  
   
 @Column(name="GT2\_OPERATING\_HOUR", precision=20, scale=10)  
 public BigDecimal getGt2OperatingHour() {  
 return this.gt2OperatingHour;  
 }  
   
 public void setGt2OperatingHour(BigDecimal gt2OperatingHour) {  
 this.gt2OperatingHour = gt2OperatingHour;  
 }  
  
   
 @Column(name="NET\_OUTPUT\_CORRECTED\_TO\_SUMMER\_DESIGN\_CFE", precision=20, scale=10)  
 public BigDecimal getNetOutputCorrectedToSummerDesignCfe() {  
 return this.netOutputCorrectedToSummerDesignCfe;  
 }  
   
 public void setNetOutputCorrectedToSummerDesignCfe(BigDecimal netOutputCorrectedToSummerDesignCfe) {  
 this.netOutputCorrectedToSummerDesignCfe = netOutputCorrectedToSummerDesignCfe;  
 }  
  
   
 @Column(name="KCACTUAL\_GAS", precision=20, scale=10)  
 public BigDecimal getKcactualGas() {  
 return this.kcactualGas;  
 }  
   
 public void setKcactualGas(BigDecimal kcactualGas) {  
 this.kcactualGas = kcactualGas;  
 }  
  
   
 @Column(name="KCACTUAL\_OIL", precision=20, scale=10)  
 public BigDecimal getKcactualOil() {  
 return this.kcactualOil;  
 }  
   
 public void setKcactualOil(BigDecimal kcactualOil) {  
 this.kcactualOil = kcactualOil;  
 }  
  
   
 @Column(name="SPARE", precision=20, scale=10)  
 public BigDecimal getSpare() {  
 return this.spare;  
 }  
   
 public void setSpare(BigDecimal spare) {  
 this.spare = spare;  
 }  
  
   
 @Column(name="GT1\_52G\_ON", precision=20, scale=10)  
 public BigDecimal getGt152gOn() {  
 return this.gt152gOn;  
 }  
   
 public void setGt152gOn(BigDecimal gt152gOn) {  
 this.gt152gOn = gt152gOn;  
 }  
  
   
 @Column(name="GT2\_52G\_ON", precision=20, scale=10)  
 public BigDecimal getGt252gOn() {  
 return this.gt252gOn;  
 }  
   
 public void setGt252gOn(BigDecimal gt252gOn) {  
 this.gt252gOn = gt252gOn;  
 }  
  
   
 @Column(name="E\_A1010\_400KV\_S\_S\_CB\_GT1\_ABN", precision=20, scale=10)  
 public BigDecimal getEA1010400kvSSCbGt1Abn() {  
 return this.EA1010400kvSSCbGt1Abn;  
 }  
   
 public void setEA1010400kvSSCbGt1Abn(BigDecimal EA1010400kvSSCbGt1Abn) {  
 this.EA1010400kvSSCbGt1Abn = EA1010400kvSSCbGt1Abn;  
 }  
  
   
 @Column(name="E\_A1010\_400KV\_S\_S\_CB\_GT1\_CLOSE", precision=20, scale=10)  
 public BigDecimal getEA1010400kvSSCbGt1Close() {  
 return this.EA1010400kvSSCbGt1Close;  
 }  
   
 public void setEA1010400kvSSCbGt1Close(BigDecimal EA1010400kvSSCbGt1Close) {  
 this.EA1010400kvSSCbGt1Close = EA1010400kvSSCbGt1Close;  
 }  
  
   
 @Column(name="E\_A1010\_400KV\_S\_S\_CB\_GT1\_OPEN", precision=20, scale=10)  
 public BigDecimal getEA1010400kvSSCbGt1Open() {  
 return this.EA1010400kvSSCbGt1Open;  
 }  
   
 public void setEA1010400kvSSCbGt1Open(BigDecimal EA1010400kvSSCbGt1Open) {  
 this.EA1010400kvSSCbGt1Open = EA1010400kvSSCbGt1Open;  
 }  
  
   
 @Column(name="E\_A1020\_400KV\_S\_S\_CB\_GT2\_ABN", precision=20, scale=10)  
 public BigDecimal getEA1020400kvSSCbGt2Abn() {  
 return this.EA1020400kvSSCbGt2Abn;  
 }  
   
 public void setEA1020400kvSSCbGt2Abn(BigDecimal EA1020400kvSSCbGt2Abn) {  
 this.EA1020400kvSSCbGt2Abn = EA1020400kvSSCbGt2Abn;  
 }  
  
   
 @Column(name="E\_A1020\_400KV\_S\_S\_CB\_GT2\_CLOSE", precision=20, scale=10)  
 public BigDecimal getEA1020400kvSSCbGt2Close() {  
 return this.EA1020400kvSSCbGt2Close;  
 }  
   
 public void setEA1020400kvSSCbGt2Close(BigDecimal EA1020400kvSSCbGt2Close) {  
 this.EA1020400kvSSCbGt2Close = EA1020400kvSSCbGt2Close;  
 }  
  
   
 @Column(name="E\_A1020\_400KV\_S\_S\_CB\_GT2\_OPEN", precision=20, scale=10)  
 public BigDecimal getEA1020400kvSSCbGt2Open() {  
 return this.EA1020400kvSSCbGt2Open;  
 }  
   
 public void setEA1020400kvSSCbGt2Open(BigDecimal EA1020400kvSSCbGt2Open) {  
 this.EA1020400kvSSCbGt2Open = EA1020400kvSSCbGt2Open;  
 }  
  
   
 @Column(name="E\_A1030\_400KV\_S\_S\_CB\_ST\_CLOSE", precision=20, scale=10)  
 public BigDecimal getEA1030400kvSSCbStClose() {  
 return this.EA1030400kvSSCbStClose;  
 }  
   
 public void setEA1030400kvSSCbStClose(BigDecimal EA1030400kvSSCbStClose) {  
 this.EA1030400kvSSCbStClose = EA1030400kvSSCbStClose;  
 }  
  
   
 @Column(name="E\_A1030\_400KV\_S\_S\_CB\_ST\_OPEN", precision=20, scale=10)  
 public BigDecimal getEA1030400kvSSCbStOpen() {  
 return this.EA1030400kvSSCbStOpen;  
 }  
   
 public void setEA1030400kvSSCbStOpen(BigDecimal EA1030400kvSSCbStOpen) {  
 this.EA1030400kvSSCbStOpen = EA1030400kvSSCbStOpen;  
 }  
  
   
 @Column(name="GT1\_SYNCHRONIZE", precision=20, scale=10)  
 public BigDecimal getGt1Synchronize() {  
 return this.gt1Synchronize;  
 }  
   
 public void setGt1Synchronize(BigDecimal gt1Synchronize) {  
 this.gt1Synchronize = gt1Synchronize;  
 }  
  
   
 @Column(name="GT2\_SYNCHRONIZE", precision=20, scale=10)  
 public BigDecimal getGt2Synchronize() {  
 return this.gt2Synchronize;  
 }  
   
 public void setGt2Synchronize(BigDecimal gt2Synchronize) {  
 this.gt2Synchronize = gt2Synchronize;  
 }  
  
   
 @Column(name="ALL\_TARIFF\_METER\_ABNORMAL", precision=20, scale=10)  
 public BigDecimal getAllTariffMeterAbnormal() {  
 return this.allTariffMeterAbnormal;  
 }  
   
 public void setAllTariffMeterAbnormal(BigDecimal allTariffMeterAbnormal) {  
 this.allTariffMeterAbnormal = allTariffMeterAbnormal;  
 }  
  
   
 @Column(name="GT1\_TRANSMITTER\_WAT\_AND\_VAR\_ABNORMAL", precision=20, scale=10)  
 public BigDecimal getGt1TransmitterWatAndVarAbnormal() {  
 return this.gt1TransmitterWatAndVarAbnormal;  
 }  
   
 public void setGt1TransmitterWatAndVarAbnormal(BigDecimal gt1TransmitterWatAndVarAbnormal) {  
 this.gt1TransmitterWatAndVarAbnormal = gt1TransmitterWatAndVarAbnormal;  
 }  
  
   
 @Column(name="GT2\_TRANSMITTER\_WAT\_AND\_VAR\_ABNORMAL", precision=20, scale=10)  
 public BigDecimal getGt2TransmitterWatAndVarAbnormal() {  
 return this.gt2TransmitterWatAndVarAbnormal;  
 }  
   
 public void setGt2TransmitterWatAndVarAbnormal(BigDecimal gt2TransmitterWatAndVarAbnormal) {  
 this.gt2TransmitterWatAndVarAbnormal = gt2TransmitterWatAndVarAbnormal;  
 }  
  
   
 @Column(name="ST\_TRANSMITTER\_WAT\_AND\_VAR\_ABNORMAL", precision=20, scale=10)  
 public BigDecimal getStTransmitterWatAndVarAbnormal() {  
 return this.stTransmitterWatAndVarAbnormal;  
 }  
   
 public void setStTransmitterWatAndVarAbnormal(BigDecimal stTransmitterWatAndVarAbnormal) {  
 this.stTransmitterWatAndVarAbnormal = stTransmitterWatAndVarAbnormal;  
 }  
  
   
 @Column(name="NO1\_TEMP\_TRANSMITTER\_ABNORMAL", precision=20, scale=10)  
 public BigDecimal getNo1TempTransmitterAbnormal() {  
 return this.no1TempTransmitterAbnormal;  
 }  
   
 public void setNo1TempTransmitterAbnormal(BigDecimal no1TempTransmitterAbnormal) {  
 this.no1TempTransmitterAbnormal = no1TempTransmitterAbnormal;  
 }  
  
   
 @Column(name="NO2\_TEMP\_TRANSMITTER\_ABNORMAL", precision=20, scale=10)  
 public BigDecimal getNo2TempTransmitterAbnormal() {  
 return this.no2TempTransmitterAbnormal;  
 }  
   
 public void setNo2TempTransmitterAbnormal(BigDecimal no2TempTransmitterAbnormal) {  
 this.no2TempTransmitterAbnormal = no2TempTransmitterAbnormal;  
 }  
  
   
 @Column(name="TEMP\_TRANSMITTER\_SELECT\_0\_NO2\_1\_NO1", precision=20, scale=10)  
 public BigDecimal getTempTransmitterSelect0No21No1() {  
 return this.tempTransmitterSelect0No21No1;  
 }  
   
 public void setTempTransmitterSelect0No21No1(BigDecimal tempTransmitterSelect0No21No1) {  
 this.tempTransmitterSelect0No21No1 = tempTransmitterSelect0No21No1;  
 }  
  
   
 @Column(name="NO1\_HUMIDITY\_TRANSMITTER\_ABNORMAL", precision=20, scale=10)  
 public BigDecimal getNo1HumidityTransmitterAbnormal() {  
 return this.no1HumidityTransmitterAbnormal;  
 }  
   
 public void setNo1HumidityTransmitterAbnormal(BigDecimal no1HumidityTransmitterAbnormal) {  
 this.no1HumidityTransmitterAbnormal = no1HumidityTransmitterAbnormal;  
 }  
  
   
 @Column(name="NO2\_HUMIDITY\_TRANSMITTER\_ABNORMAL", precision=20, scale=10)  
 public BigDecimal getNo2HumidityTransmitterAbnormal() {  
 return this.no2HumidityTransmitterAbnormal;  
 }  
   
 public void setNo2HumidityTransmitterAbnormal(BigDecimal no2HumidityTransmitterAbnormal) {  
 this.no2HumidityTransmitterAbnormal = no2HumidityTransmitterAbnormal;  
 }  
  
   
 @Column(name="HUMIDITY\_TRANSMITTER\_SELECT\_0\_NO2\_1\_NO1", precision=20, scale=10)  
 public BigDecimal getHumidityTransmitterSelect0No21No1() {  
 return this.humidityTransmitterSelect0No21No1;  
 }  
   
 public void setHumidityTransmitterSelect0No21No1(BigDecimal humidityTransmitterSelect0No21No1) {  
 this.humidityTransmitterSelect0No21No1 = humidityTransmitterSelect0No21No1;  
 }  
  
   
 @Column(name="NO1\_AMBIENT\_PRESSURE\_TRANSMITTER\_ABNORMAL", precision=20, scale=10)  
 public BigDecimal getNo1AmbientPressureTransmitterAbnormal() {  
 return this.no1AmbientPressureTransmitterAbnormal;  
 }  
   
 public void setNo1AmbientPressureTransmitterAbnormal(BigDecimal no1AmbientPressureTransmitterAbnormal) {  
 this.no1AmbientPressureTransmitterAbnormal = no1AmbientPressureTransmitterAbnormal;  
 }  
  
   
 @Column(name="NO2\_AMBIENT\_PRESSURE\_TRANSMITTER\_ABNORMAL", precision=20, scale=10)  
 public BigDecimal getNo2AmbientPressureTransmitterAbnormal() {  
 return this.no2AmbientPressureTransmitterAbnormal;  
 }  
   
 public void setNo2AmbientPressureTransmitterAbnormal(BigDecimal no2AmbientPressureTransmitterAbnormal) {  
 this.no2AmbientPressureTransmitterAbnormal = no2AmbientPressureTransmitterAbnormal;  
 }  
  
   
 @Column(name="AMBIENT\_PRESSURE\_TRANSMITTER\_SELECT\_0\_NO2\_1\_NO1", precision=20, scale=10)  
 public BigDecimal getAmbientPressureTransmitterSelect0No21No1() {  
 return this.ambientPressureTransmitterSelect0No21No1;  
 }  
   
 public void setAmbientPressureTransmitterSelect0No21No1(BigDecimal ambientPressureTransmitterSelect0No21No1) {  
 this.ambientPressureTransmitterSelect0No21No1 = ambientPressureTransmitterSelect0No21No1;  
 }  
  
   
 @Column(name="NO1\_COND\_VACUUM\_TRANSMITTER\_ABNORMAL", precision=20, scale=10)  
 public BigDecimal getNo1CondVacuumTransmitterAbnormal() {  
 return this.no1CondVacuumTransmitterAbnormal;  
 }  
   
 public void setNo1CondVacuumTransmitterAbnormal(BigDecimal no1CondVacuumTransmitterAbnormal) {  
 this.no1CondVacuumTransmitterAbnormal = no1CondVacuumTransmitterAbnormal;  
 }  
  
   
 @Column(name="NO2\_COND\_VACUUM\_TRANSMITTER\_ABNORMAL", precision=20, scale=10)  
 public BigDecimal getNo2CondVacuumTransmitterAbnormal() {  
 return this.no2CondVacuumTransmitterAbnormal;  
 }  
   
 public void setNo2CondVacuumTransmitterAbnormal(BigDecimal no2CondVacuumTransmitterAbnormal) {  
 this.no2CondVacuumTransmitterAbnormal = no2CondVacuumTransmitterAbnormal;  
 }  
  
   
 @Column(name="COND\_VACUUM\_TRANSMITTER\_SELECT\_0\_NO2\_1\_NO1", precision=20, scale=10)  
 public BigDecimal getCondVacuumTransmitterSelect0No21No1() {  
 return this.condVacuumTransmitterSelect0No21No1;  
 }  
   
 public void setCondVacuumTransmitterSelect0No21No1(BigDecimal condVacuumTransmitterSelect0No21No1) {  
 this.condVacuumTransmitterSelect0No21No1 = condVacuumTransmitterSelect0No21No1;  
 }  
  
   
 @Column(name="GT1\_GAS\_SELECT", precision=20, scale=10)  
 public BigDecimal getGt1GasSelect() {  
 return this.gt1GasSelect;  
 }  
   
 public void setGt1GasSelect(BigDecimal gt1GasSelect) {  
 this.gt1GasSelect = gt1GasSelect;  
 }  
  
   
 @Column(name="GT2\_GAS\_SELECT", precision=20, scale=10)  
 public BigDecimal getGt2GasSelect() {  
 return this.gt2GasSelect;  
 }  
   
 public void setGt2GasSelect(BigDecimal gt2GasSelect) {  
 this.gt2GasSelect = gt2GasSelect;  
 }  
  
   
 @Column(name="GT1\_OIL\_SELECT", precision=20, scale=10)  
 public BigDecimal getGt1OilSelect() {  
 return this.gt1OilSelect;  
 }  
   
 public void setGt1OilSelect(BigDecimal gt1OilSelect) {  
 this.gt1OilSelect = gt1OilSelect;  
 }  
  
   
 @Column(name="GT2\_OIL\_SELECT", precision=20, scale=10)  
 public BigDecimal getGt2OilSelect() {  
 return this.gt2OilSelect;  
 }  
   
 public void setGt2OilSelect(BigDecimal gt2OilSelect) {  
 this.gt2OilSelect = gt2OilSelect;  
 }  
  
   
 @Column(name="NO1\_GT\_FCB\_SELECT", precision=20, scale=10)  
 public BigDecimal getNo1GtFcbSelect() {  
 return this.no1GtFcbSelect;  
 }  
   
 public void setNo1GtFcbSelect(BigDecimal no1GtFcbSelect) {  
 this.no1GtFcbSelect = no1GtFcbSelect;  
 }  
  
   
 @Column(name="NO2\_GT\_FCB\_SELECT", precision=20, scale=10)  
 public BigDecimal getNo2GtFcbSelect() {  
 return this.no2GtFcbSelect;  
 }  
   
 public void setNo2GtFcbSelect(BigDecimal no2GtFcbSelect) {  
 this.no2GtFcbSelect = no2GtFcbSelect;  
 }  
  
   
 @Column(name="FCB\_COMMAND", precision=20, scale=10)  
 public BigDecimal getFcbCommand() {  
 return this.fcbCommand;  
 }  
   
 public void setFcbCommand(BigDecimal fcbCommand) {  
 this.fcbCommand = fcbCommand;  
 }  
  
   
 @Column(name="GT1\_GAS\_COMB", precision=20, scale=10)  
 public BigDecimal getGt1GasComb() {  
 return this.gt1GasComb;  
 }  
   
 public void setGt1GasComb(BigDecimal gt1GasComb) {  
 this.gt1GasComb = gt1GasComb;  
 }  
  
   
 @Column(name="GT2\_GAS\_COMB", precision=20, scale=10)  
 public BigDecimal getGt2GasComb() {  
 return this.gt2GasComb;  
 }  
   
 public void setGt2GasComb(BigDecimal gt2GasComb) {  
 this.gt2GasComb = gt2GasComb;  
 }  
  
   
 @Column(name="GT1\_OIL\_COMB", precision=20, scale=10)  
 public BigDecimal getGt1OilComb() {  
 return this.gt1OilComb;  
 }  
   
 public void setGt1OilComb(BigDecimal gt1OilComb) {  
 this.gt1OilComb = gt1OilComb;  
 }  
  
   
 @Column(name="GT2\_OIL\_COMB", precision=20, scale=10)  
 public BigDecimal getGt2OilComb() {  
 return this.gt2OilComb;  
 }  
   
 public void setGt2OilComb(BigDecimal gt2OilComb) {  
 this.gt2OilComb = gt2OilComb;  
 }  
  
   
 @Column(name="GT1\_ENCENDER\_COMB", precision=20, scale=10)  
 public BigDecimal getGt1EncenderComb() {  
 return this.gt1EncenderComb;  
 }  
   
 public void setGt1EncenderComb(BigDecimal gt1EncenderComb) {  
 this.gt1EncenderComb = gt1EncenderComb;  
 }  
  
   
 @Column(name="GT2\_ENCENDER\_COMB", precision=20, scale=10)  
 public BigDecimal getGt2EncenderComb() {  
 return this.gt2EncenderComb;  
 }  
   
 public void setGt2EncenderComb(BigDecimal gt2EncenderComb) {  
 this.gt2EncenderComb = gt2EncenderComb;  
 }  
  
   
 @Column(name="ACTUAL\_HIGHER\_HEATING\_VALUE\_WET", precision=20, scale=10)  
 public BigDecimal getActualHigherHeatingValueWet() {  
 return this.actualHigherHeatingValueWet;  
 }  
   
 public void setActualHigherHeatingValueWet(BigDecimal actualHigherHeatingValueWet) {  
 this.actualHigherHeatingValueWet = actualHigherHeatingValueWet;  
 }  
  
   
 @Column(name="ACTUAL\_LOWER\_HEATING\_VALUE\_WET", precision=20, scale=10)  
 public BigDecimal getActualLowerHeatingValueWet() {  
 return this.actualLowerHeatingValueWet;  
 }  
   
 public void setActualLowerHeatingValueWet(BigDecimal actualLowerHeatingValueWet) {  
 this.actualLowerHeatingValueWet = actualLowerHeatingValueWet;  
 }  
  
   
 @Column(name="ACTUAL\_HIGHER\_HEATING\_VALUE\_WET2", precision=20, scale=10)  
 public BigDecimal getActualHigherHeatingValueWet2() {  
 return this.actualHigherHeatingValueWet2;  
 }  
   
 public void setActualHigherHeatingValueWet2(BigDecimal actualHigherHeatingValueWet2) {  
 this.actualHigherHeatingValueWet2 = actualHigherHeatingValueWet2;  
 }  
  
   
 @Column(name="ACTUAL\_LOWER\_HEATING\_VALUE\_WET2", precision=20, scale=10)  
 public BigDecimal getActualLowerHeatingValueWet2() {  
 return this.actualLowerHeatingValueWet2;  
 }  
   
 public void setActualLowerHeatingValueWet2(BigDecimal actualLowerHeatingValueWet2) {  
 this.actualLowerHeatingValueWet2 = actualLowerHeatingValueWet2;  
 }  
  
   
 @Column(name="GAS\_VOLUME\_TOTAL", precision=20, scale=10)  
 public BigDecimal getGasVolumeTotal() {  
 return this.gasVolumeTotal;  
 }  
   
 public void setGasVolumeTotal(BigDecimal gasVolumeTotal) {  
 this.gasVolumeTotal = gasVolumeTotal;  
 }  
  
   
 @Column(name="GAS\_MASS\_TOTAL", precision=20, scale=10)  
 public BigDecimal getGasMassTotal() {  
 return this.gasMassTotal;  
 }  
   
 public void setGasMassTotal(BigDecimal gasMassTotal) {  
 this.gasMassTotal = gasMassTotal;  
 }  
  
   
 @Column(name="QG5\_5MIN\_HEAT\_INPUT\_WITH\_GAS", precision=20, scale=10)  
 public BigDecimal getQg55minHeatInputWithGas() {  
 return this.qg55minHeatInputWithGas;  
 }  
   
 public void setQg55minHeatInputWithGas(BigDecimal qg55minHeatInputWithGas) {  
 this.qg55minHeatInputWithGas = qg55minHeatInputWithGas;  
 }  
  
   
 @Column(name="QD5\_5MIN\_HEAT\_INPUT\_WITH\_OIL", precision=20, scale=10)  
 public BigDecimal getQd55minHeatInputWithOil() {  
 return this.qd55minHeatInputWithOil;  
 }  
   
 public void setQd55minHeatInputWithOil(BigDecimal qd55minHeatInputWithOil) {  
 this.qd55minHeatInputWithOil = qd55minHeatInputWithOil;  
 }  
  
   
 @Column(name="GT1\_OPERATION\_MODE", precision=20, scale=10)  
 public BigDecimal getGt1OperationMode() {  
 return this.gt1OperationMode;  
 }  
   
 public void setGt1OperationMode(BigDecimal gt1OperationMode) {  
 this.gt1OperationMode = gt1OperationMode;  
 }  
  
   
 @Column(name="GT2\_OPERATION\_MODE", precision=20, scale=10)  
 public BigDecimal getGt2OperationMode() {  
 return this.gt2OperationMode;  
 }  
   
 public void setGt2OperationMode(BigDecimal gt2OperationMode) {  
 this.gt2OperationMode = gt2OperationMode;  
 }  
  
   
 @Column(name="SPARE1", precision=20, scale=10)  
 public BigDecimal getSpare1() {  
 return this.spare1;  
 }  
   
 public void setSpare1(BigDecimal spare1) {  
 this.spare1 = spare1;  
 }  
  
   
 @Column(name="SPARE2", precision=20, scale=10)  
 public BigDecimal getSpare2() {  
 return this.spare2;  
 }  
   
 public void setSpare2(BigDecimal spare2) {  
 this.spare2 = spare2;  
 }  
  
   
 @Column(name="SPARE3", precision=20, scale=10)  
 public BigDecimal getSpare3() {  
 return this.spare3;  
 }  
   
 public void setSpare3(BigDecimal spare3) {  
 this.spare3 = spare3;  
 }  
  
   
 @Column(name="TIPO\_CARGA", length=20)  
 public String getTipoCarga() {  
 return this.tipoCarga;  
 }  
   
 public void setTipoCarga(String tipoCarga) {  
 this.tipoCarga = tipoCarga;  
 }  
  
   
 @Column(name="CARGA\_ID", precision=22, scale=0)  
 public BigDecimal getCargaId() {  
 return this.cargaId;  
 }  
   
 public void setCargaId(BigDecimal cargaId) {  
 this.cargaId = cargaId;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="HORA\_CARGA", length=11)  
 public Date getHoraCarga() {  
 return this.horaCarga;  
 }  
   
 public void setHoraCarga(Date horaCarga) {  
 this.horaCarga = horaCarga;  
 }  
  
   
 @Column(name="ESTATUS\_LIMPIO", precision=22, scale=0)  
 public BigDecimal getEstatusLimpio() {  
 return this.estatusLimpio;  
 }  
   
 public void setEstatusLimpio(BigDecimal estatusLimpio) {  
 this.estatusLimpio = estatusLimpio;  
 }  
  
   
 @Column(name="CRUDOS\_ID", precision=22, scale=0)  
 public BigDecimal getCrudosId() {  
 return this.crudosId;  
 }  
   
 public void setCrudosId(BigDecimal crudosId) {  
 this.crudosId = crudosId;  
 }  
  
   
 @Column(name="CORRECTO", precision=1, scale=0)  
 public Boolean getCorrecto() {  
 return this.correcto;  
 }  
   
 public void setCorrecto(Boolean correcto) {  
 this.correcto = correcto;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxCrudosSol.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.Date;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* TxCrudosSol generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_CRUDOS\_SOL"  
)  
public class TxCrudosSol implements java.io.Serializable {  
  
  
 private BigDecimal crudosId;  
 private String dia;  
 private String minuto;  
 private String fuelSelect;  
 private String hourlyLoadInUse;  
 private String line1WattHourMas5min;  
 private String line2WattHourMas5min;  
 private String line1WattHourMenos5min;  
 private String line2WattHourMenos5min;  
 private String line1WattHourMas5minBackup;  
 private String line2WattHourMas5minBackup;  
 private String line1WattHourMenos5minBackup;  
 private String line2WattHourMenos5minBackup;  
 private String gt1ActivePower;  
 private String gt2ActivePower;  
 private String stActivePower;  
 private String gt1ReactivePower;  
 private String gt2ReactivePower;  
 private String stReactivePower;  
 private String line1WattInstant;  
 private String line2WattInstant;  
 private String line1VarInstant;  
 private String line2VarInstant;  
 private String plantNetWattHourCalculationFromGross;  
 private String stgWattHour;  
 private String stgVarHour;  
 private String no1GtgWattHour;  
 private String no1GtgVarHour;  
 private String no1GtgUnitHouseLoad;  
 private String no2GtgWattHour;  
 private String no2GtgVarHour;  
 private String no2GtgUnitHouseLoad;  
 private String ambientTemperatureInUse;  
 private String ambientTemperature1;  
 private String ambientTemperature2;  
 private String relativeHumidityInUse;  
 private String relativeHumidity1;  
 private String relativeHumidity2;  
 private String ambientPressureInUse;  
 private String ambientPressure1;  
 private String ambientPressure2;  
 private String seaWaterTemperatureInUse;  
 private String seaWaterTemperature1;  
 private String seaWaterTemperature2;  
 private String spare;  
 private String fuelGasHigherHeatingValue;  
 private String fuelGasLowerHeatingValue;  
 private String fuelGasHigherHeatingValue2;  
 private String fuelGasLowerHeatingValue2;  
 private String fuelOilHigherHeatingValeu;  
 private String fuelOilLowerHeatingValue;  
 private String fuelOilHigherHeatingValeu2;  
 private String fuelOilLowerHeatingValue2;  
 private String fuelOilSpecificGravity;  
 private String fuelGasFlowCurrentMasFlow;  
 private String strm1CurrentVolumeFlowRate;  
 private String fuelGasFlow5minAccumulatedValue;  
 private String oilFlowGt1Gt2;  
 private String gt1OilFlowCurrentValue;  
 private String gt2OilFlowCurrentValue;  
 private String oilFlowGt1Gt22;  
 private String gt1OilFlow5minAccumulatedValue;  
 private String gt2OilFlow5minAccumulatedValue;  
 private String overallPowerFactor;  
 private String netPowerOf5min5minAccumulatedWithGasFiring;  
 private String netPowerOf5min5minAccumulatedWithOilFiring;  
 private String loadPercentageAtGas;  
 private String loadPercentageAtOil;  
 private String ctungAtYearAverageDesignConditionGas;  
 private String ctungAtYearAverageDesignConditionOil;  
 private String correctedCtungGas;  
 private String correctedCtungOil;  
 private String fuelGasPressure;  
 private String gasTemperature;  
 private String loadDemand;  
 private String eh512RollingAverage;  
 private String ambientTemperature12RollingAverage;  
 private String relativeHumidity12RollingAverage;  
 private String ambientPressure12RollingAverage;  
 private String condenserVacuum12RollingAverage;  
 private String fuelGasFlow12RollingAverage;  
 private String fuelGasFlow12RollingAverage2;  
 private String fuelGasPressure12RollingAverage;  
 private String overallPowerFactor12RollingAverage;  
 private String fuelOilFlow12RollingAverage;  
 private String fuelOilFlow12RollingAverage2;  
 private String correctedCtung12RollingWeightedAverageGas;  
 private String correctedCtung12RollingWeightedAverageOil;  
 private String fuelGasHigherHeatingValue12RollingAverage;  
 private String fuelGasLowerHeatingValue12RollingAverage;  
 private String fuelGasHigherHeatingValue12RollingAverage2;  
 private String fuelGasLowerHeatingValue12RollingAverage2;  
 private String fuelOilHigherHeatingValue12RollingAverage;  
 private String fuelOilLowerHeatingValue12RollingAverage;  
 private String fuelOilHigherHeatingValue12RollingAverage2;  
 private String fuelOilLowerHeatingValue12RollingAverage2;  
 private String numberOfMinutesOfOperatingAtGasFiring;  
 private String numberOfMinutesOfOperatingAtOilFiring;  
 private String eh12RollingAverage;  
 private String egh12RollingAverage;  
 private String edh12RollingAverage;  
 private String nhovgOperatingHourAtNoLoadBurningGas;  
 private String nhovdOperatingHourAtNoLoadBurningOil;  
 private String gt1OperatingHour;  
 private String gt2OperatingHour;  
 private String netOutputCorrectedToSummerDesignCfe;  
 private String cnGas;  
 private String cnOil;  
 private String spare2;  
 private String gt152gOn;  
 private String gt252gOn;  
 private String EA1010400kvSSCbGt1Abn;  
 private String EA1010400kvSSCbGt1Close;  
 private String EA1010400kvSSCbGt1Open;  
 private String EA1020400kvSSCbGt2Abn;  
 private String EA1020400kvSSCbGt2Close;  
 private String EA1020400kvSSCbGt2Open;  
 private String EA1030400kvSSCbStClose;  
 private String EA1030400kvSSCbStOpen;  
 private String gt1Synchronize;  
 private String gt2Synchronize;  
 private String allTariffMeterAbnormal;  
 private String gt1TransmitterWatAndVarAbnormal;  
 private String gt2TransmitterWatAndVarAbnormal;  
 private String stTransmitterWatAndVarAbnormal;  
 private String no1TempTransmitterAbnormal;  
 private String no2TempTransmitterAbnormal;  
 private String tempTransmitterSelect0No21No1;  
 private String no1HumidityTransmitterAbnormal;  
 private String no2HumidityTransmitterAbnormal;  
 private String humidityTransmitterSelect0No21No1;  
 private String no1AmbientPressureTransmitterAbnormal;  
 private String no2AmbientPressureTransmitterAbnormal;  
 private String ambientPressureTransmitterSelect0No21No1;  
 private String no1SeaWaterTemperatureTransmitterAbnormal;  
 private String no2SeaWaterTemperatureTransmitterAbnormal;  
 private String seaWaterTemperatureTransmitterSelect0No21No1;  
 private String gt1GasSelect;  
 private String gt2GasSelect;  
 private String gt1OilSelect;  
 private String gt2OilSelect;  
 private String no1GtFcbSelect;  
 private String no2GtFcbSelect;  
 private String fcbCommand;  
 private String gt1GasComb;  
 private String gt2GasComb;  
 private String gt1OilComb;  
 private String gt2OilComb;  
 private String gt1EncenderComb;  
 private String gt2EncenderComb;  
 private String actualHigherHeatingValueWet;  
 private String actualLowerHeatingValueWet;  
 private String actualHigherHeatingValueWet2;  
 private String actualLowerHeatingValueWet2;  
 private String gasVolumeTotal;  
 private String gasMassTotal;  
 private String qg55minHeatInputWithGas;  
 private String qd55minHeatInputWithOil;  
 private String gt1OperationMode;  
 private String gt2OperationMode;  
 private String line1WattHourMas5minInUse;  
 private String line2WattHourMas5minInUse;  
 private String ctov;  
 private String line1VarHour5minKvarh;  
 private String line2VarHour5minKvarh;  
 private String line1VarHour5minKvarhBackup;  
 private String line2VarHour5minKvarhBackup;  
 private String cn;  
 private String cng100corrCng;  
 private String line1WattHourMenos5minInUse;  
 private String line2WattHourMenos5minInUse;  
 private String kilovarhourPrevious;  
 private String line1KilovarhourPreviousProgq1InUse;  
 private String line1KilovarhourPreviousProgq4InUse;  
 private String line2KilovarhourPreviousProgq1InUse;  
 private String line2KilovarhourPreviousProgq4InUse;  
 private String line1KilovarhourPreviousProgq1Main;  
 private String line1KilovarhourPreviousProgq4Main;  
 private String line2KilovarhourPreviousProgq1Main;  
 private String line2KilovarhourPreviousProgq4Main;  
 private String line1KilovarhourPreviousProgq1Backup;  
 private String line1KilovarhourPreviousProgq4Backup;  
 private String line2KilovarhourPreviousProgq1Backup;  
 private String line2KilovarhourPreviousProgq4Backup;  
 private String tipoCarga;  
 private BigDecimal cargaId;  
 private Date horaCarga;  
 private BigDecimal estatusLimpio;  
  
 public TxCrudosSol() {  
 }  
  
   
 public TxCrudosSol(Date horaCarga, BigDecimal estatusLimpio) {  
 this.horaCarga = horaCarga;  
 this.estatusLimpio = estatusLimpio;  
 }  
 public TxCrudosSol(String dia, String minuto, String fuelSelect, String hourlyLoadInUse, String line1WattHourMas5min, String line2WattHourMas5min, String line1WattHourMenos5min, String line2WattHourMenos5min, String line1WattHourMas5minBackup, String line2WattHourMas5minBackup, String line1WattHourMenos5minBackup, String line2WattHourMenos5minBackup, String gt1ActivePower, String gt2ActivePower, String stActivePower, String gt1ReactivePower, String gt2ReactivePower, String stReactivePower, String line1WattInstant, String line2WattInstant, String line1VarInstant, String line2VarInstant, String plantNetWattHourCalculationFromGross, String stgWattHour, String stgVarHour, String no1GtgWattHour, String no1GtgVarHour, String no1GtgUnitHouseLoad, String no2GtgWattHour, String no2GtgVarHour, String no2GtgUnitHouseLoad, String ambientTemperatureInUse, String ambientTemperature1, String ambientTemperature2, String relativeHumidityInUse, String relativeHumidity1, String relativeHumidity2, String ambientPressureInUse, String ambientPressure1, String ambientPressure2, String seaWaterTemperatureInUse, String seaWaterTemperature1, String seaWaterTemperature2, String spare, String fuelGasHigherHeatingValue, String fuelGasLowerHeatingValue, String fuelGasHigherHeatingValue2, String fuelGasLowerHeatingValue2, String fuelOilHigherHeatingValeu, String fuelOilLowerHeatingValue, String fuelOilHigherHeatingValeu2, String fuelOilLowerHeatingValue2, String fuelOilSpecificGravity, String fuelGasFlowCurrentMasFlow, String strm1CurrentVolumeFlowRate, String fuelGasFlow5minAccumulatedValue, String oilFlowGt1Gt2, String gt1OilFlowCurrentValue, String gt2OilFlowCurrentValue, String oilFlowGt1Gt22, String gt1OilFlow5minAccumulatedValue, String gt2OilFlow5minAccumulatedValue, String overallPowerFactor, String netPowerOf5min5minAccumulatedWithGasFiring, String netPowerOf5min5minAccumulatedWithOilFiring, String loadPercentageAtGas, String loadPercentageAtOil, String ctungAtYearAverageDesignConditionGas, String ctungAtYearAverageDesignConditionOil, String correctedCtungGas, String correctedCtungOil, String fuelGasPressure, String gasTemperature, String loadDemand, String eh512RollingAverage, String ambientTemperature12RollingAverage, String relativeHumidity12RollingAverage, String ambientPressure12RollingAverage, String condenserVacuum12RollingAverage, String fuelGasFlow12RollingAverage, String fuelGasFlow12RollingAverage2, String fuelGasPressure12RollingAverage, String overallPowerFactor12RollingAverage, String fuelOilFlow12RollingAverage, String fuelOilFlow12RollingAverage2, String correctedCtung12RollingWeightedAverageGas, String correctedCtung12RollingWeightedAverageOil, String fuelGasHigherHeatingValue12RollingAverage, String fuelGasLowerHeatingValue12RollingAverage, String fuelGasHigherHeatingValue12RollingAverage2, String fuelGasLowerHeatingValue12RollingAverage2, String fuelOilHigherHeatingValue12RollingAverage, String fuelOilLowerHeatingValue12RollingAverage, String fuelOilHigherHeatingValue12RollingAverage2, String fuelOilLowerHeatingValue12RollingAverage2, String numberOfMinutesOfOperatingAtGasFiring, String numberOfMinutesOfOperatingAtOilFiring, String eh12RollingAverage, String egh12RollingAverage, String edh12RollingAverage, String nhovgOperatingHourAtNoLoadBurningGas, String nhovdOperatingHourAtNoLoadBurningOil, String gt1OperatingHour, String gt2OperatingHour, String netOutputCorrectedToSummerDesignCfe, String cnGas, String cnOil, String spare2, String gt152gOn, String gt252gOn, String EA1010400kvSSCbGt1Abn, String EA1010400kvSSCbGt1Close, String EA1010400kvSSCbGt1Open, String EA1020400kvSSCbGt2Abn, String EA1020400kvSSCbGt2Close, String EA1020400kvSSCbGt2Open, String EA1030400kvSSCbStClose, String EA1030400kvSSCbStOpen, String gt1Synchronize, String gt2Synchronize, String allTariffMeterAbnormal, String gt1TransmitterWatAndVarAbnormal, String gt2TransmitterWatAndVarAbnormal, String stTransmitterWatAndVarAbnormal, String no1TempTransmitterAbnormal, String no2TempTransmitterAbnormal, String tempTransmitterSelect0No21No1, String no1HumidityTransmitterAbnormal, String no2HumidityTransmitterAbnormal, String humidityTransmitterSelect0No21No1, String no1AmbientPressureTransmitterAbnormal, String no2AmbientPressureTransmitterAbnormal, String ambientPressureTransmitterSelect0No21No1, String no1SeaWaterTemperatureTransmitterAbnormal, String no2SeaWaterTemperatureTransmitterAbnormal, String seaWaterTemperatureTransmitterSelect0No21No1, String gt1GasSelect, String gt2GasSelect, String gt1OilSelect, String gt2OilSelect, String no1GtFcbSelect, String no2GtFcbSelect, String fcbCommand, String gt1GasComb, String gt2GasComb, String gt1OilComb, String gt2OilComb, String gt1EncenderComb, String gt2EncenderComb, String actualHigherHeatingValueWet, String actualLowerHeatingValueWet, String actualHigherHeatingValueWet2, String actualLowerHeatingValueWet2, String gasVolumeTotal, String gasMassTotal, String qg55minHeatInputWithGas, String qd55minHeatInputWithOil, String gt1OperationMode, String gt2OperationMode, String line1WattHourMas5minInUse, String line2WattHourMas5minInUse, String ctov, String line1VarHour5minKvarh, String line2VarHour5minKvarh, String line1VarHour5minKvarhBackup, String line2VarHour5minKvarhBackup, String cn, String cng100corrCng, String line1WattHourMenos5minInUse, String line2WattHourMenos5minInUse, String kilovarhourPrevious, String line1KilovarhourPreviousProgq1InUse, String line1KilovarhourPreviousProgq4InUse, String line2KilovarhourPreviousProgq1InUse, String line2KilovarhourPreviousProgq4InUse, String line1KilovarhourPreviousProgq1Main, String line1KilovarhourPreviousProgq4Main, String line2KilovarhourPreviousProgq1Main, String line2KilovarhourPreviousProgq4Main, String line1KilovarhourPreviousProgq1Backup, String line1KilovarhourPreviousProgq4Backup, String line2KilovarhourPreviousProgq1Backup, String line2KilovarhourPreviousProgq4Backup, String tipoCarga, BigDecimal cargaId, Date horaCarga, BigDecimal estatusLimpio) {  
 this.dia = dia;  
 this.minuto = minuto;  
 this.fuelSelect = fuelSelect;  
 this.hourlyLoadInUse = hourlyLoadInUse;  
 this.line1WattHourMas5min = line1WattHourMas5min;  
 this.line2WattHourMas5min = line2WattHourMas5min;  
 this.line1WattHourMenos5min = line1WattHourMenos5min;  
 this.line2WattHourMenos5min = line2WattHourMenos5min;  
 this.line1WattHourMas5minBackup = line1WattHourMas5minBackup;  
 this.line2WattHourMas5minBackup = line2WattHourMas5minBackup;  
 this.line1WattHourMenos5minBackup = line1WattHourMenos5minBackup;  
 this.line2WattHourMenos5minBackup = line2WattHourMenos5minBackup;  
 this.gt1ActivePower = gt1ActivePower;  
 this.gt2ActivePower = gt2ActivePower;  
 this.stActivePower = stActivePower;  
 this.gt1ReactivePower = gt1ReactivePower;  
 this.gt2ReactivePower = gt2ReactivePower;  
 this.stReactivePower = stReactivePower;  
 this.line1WattInstant = line1WattInstant;  
 this.line2WattInstant = line2WattInstant;  
 this.line1VarInstant = line1VarInstant;  
 this.line2VarInstant = line2VarInstant;  
 this.plantNetWattHourCalculationFromGross = plantNetWattHourCalculationFromGross;  
 this.stgWattHour = stgWattHour;  
 this.stgVarHour = stgVarHour;  
 this.no1GtgWattHour = no1GtgWattHour;  
 this.no1GtgVarHour = no1GtgVarHour;  
 this.no1GtgUnitHouseLoad = no1GtgUnitHouseLoad;  
 this.no2GtgWattHour = no2GtgWattHour;  
 this.no2GtgVarHour = no2GtgVarHour;  
 this.no2GtgUnitHouseLoad = no2GtgUnitHouseLoad;  
 this.ambientTemperatureInUse = ambientTemperatureInUse;  
 this.ambientTemperature1 = ambientTemperature1;  
 this.ambientTemperature2 = ambientTemperature2;  
 this.relativeHumidityInUse = relativeHumidityInUse;  
 this.relativeHumidity1 = relativeHumidity1;  
 this.relativeHumidity2 = relativeHumidity2;  
 this.ambientPressureInUse = ambientPressureInUse;  
 this.ambientPressure1 = ambientPressure1;  
 this.ambientPressure2 = ambientPressure2;  
 this.seaWaterTemperatureInUse = seaWaterTemperatureInUse;  
 this.seaWaterTemperature1 = seaWaterTemperature1;  
 this.seaWaterTemperature2 = seaWaterTemperature2;  
 this.spare = spare;  
 this.fuelGasHigherHeatingValue = fuelGasHigherHeatingValue;  
 this.fuelGasLowerHeatingValue = fuelGasLowerHeatingValue;  
 this.fuelGasHigherHeatingValue2 = fuelGasHigherHeatingValue2;  
 this.fuelGasLowerHeatingValue2 = fuelGasLowerHeatingValue2;  
 this.fuelOilHigherHeatingValeu = fuelOilHigherHeatingValeu;  
 this.fuelOilLowerHeatingValue = fuelOilLowerHeatingValue;  
 this.fuelOilHigherHeatingValeu2 = fuelOilHigherHeatingValeu2;  
 this.fuelOilLowerHeatingValue2 = fuelOilLowerHeatingValue2;  
 this.fuelOilSpecificGravity = fuelOilSpecificGravity;  
 this.fuelGasFlowCurrentMasFlow = fuelGasFlowCurrentMasFlow;  
 this.strm1CurrentVolumeFlowRate = strm1CurrentVolumeFlowRate;  
 this.fuelGasFlow5minAccumulatedValue = fuelGasFlow5minAccumulatedValue;  
 this.oilFlowGt1Gt2 = oilFlowGt1Gt2;  
 this.gt1OilFlowCurrentValue = gt1OilFlowCurrentValue;  
 this.gt2OilFlowCurrentValue = gt2OilFlowCurrentValue;  
 this.oilFlowGt1Gt22 = oilFlowGt1Gt22;  
 this.gt1OilFlow5minAccumulatedValue = gt1OilFlow5minAccumulatedValue;  
 this.gt2OilFlow5minAccumulatedValue = gt2OilFlow5minAccumulatedValue;  
 this.overallPowerFactor = overallPowerFactor;  
 this.netPowerOf5min5minAccumulatedWithGasFiring = netPowerOf5min5minAccumulatedWithGasFiring;  
 this.netPowerOf5min5minAccumulatedWithOilFiring = netPowerOf5min5minAccumulatedWithOilFiring;  
 this.loadPercentageAtGas = loadPercentageAtGas;  
 this.loadPercentageAtOil = loadPercentageAtOil;  
 this.ctungAtYearAverageDesignConditionGas = ctungAtYearAverageDesignConditionGas;  
 this.ctungAtYearAverageDesignConditionOil = ctungAtYearAverageDesignConditionOil;  
 this.correctedCtungGas = correctedCtungGas;  
 this.correctedCtungOil = correctedCtungOil;  
 this.fuelGasPressure = fuelGasPressure;  
 this.gasTemperature = gasTemperature;  
 this.loadDemand = loadDemand;  
 this.eh512RollingAverage = eh512RollingAverage;  
 this.ambientTemperature12RollingAverage = ambientTemperature12RollingAverage;  
 this.relativeHumidity12RollingAverage = relativeHumidity12RollingAverage;  
 this.ambientPressure12RollingAverage = ambientPressure12RollingAverage;  
 this.condenserVacuum12RollingAverage = condenserVacuum12RollingAverage;  
 this.fuelGasFlow12RollingAverage = fuelGasFlow12RollingAverage;  
 this.fuelGasFlow12RollingAverage2 = fuelGasFlow12RollingAverage2;  
 this.fuelGasPressure12RollingAverage = fuelGasPressure12RollingAverage;  
 this.overallPowerFactor12RollingAverage = overallPowerFactor12RollingAverage;  
 this.fuelOilFlow12RollingAverage = fuelOilFlow12RollingAverage;  
 this.fuelOilFlow12RollingAverage2 = fuelOilFlow12RollingAverage2;  
 this.correctedCtung12RollingWeightedAverageGas = correctedCtung12RollingWeightedAverageGas;  
 this.correctedCtung12RollingWeightedAverageOil = correctedCtung12RollingWeightedAverageOil;  
 this.fuelGasHigherHeatingValue12RollingAverage = fuelGasHigherHeatingValue12RollingAverage;  
 this.fuelGasLowerHeatingValue12RollingAverage = fuelGasLowerHeatingValue12RollingAverage;  
 this.fuelGasHigherHeatingValue12RollingAverage2 = fuelGasHigherHeatingValue12RollingAverage2;  
 this.fuelGasLowerHeatingValue12RollingAverage2 = fuelGasLowerHeatingValue12RollingAverage2;  
 this.fuelOilHigherHeatingValue12RollingAverage = fuelOilHigherHeatingValue12RollingAverage;  
 this.fuelOilLowerHeatingValue12RollingAverage = fuelOilLowerHeatingValue12RollingAverage;  
 this.fuelOilHigherHeatingValue12RollingAverage2 = fuelOilHigherHeatingValue12RollingAverage2;  
 this.fuelOilLowerHeatingValue12RollingAverage2 = fuelOilLowerHeatingValue12RollingAverage2;  
 this.numberOfMinutesOfOperatingAtGasFiring = numberOfMinutesOfOperatingAtGasFiring;  
 this.numberOfMinutesOfOperatingAtOilFiring = numberOfMinutesOfOperatingAtOilFiring;  
 this.eh12RollingAverage = eh12RollingAverage;  
 this.egh12RollingAverage = egh12RollingAverage;  
 this.edh12RollingAverage = edh12RollingAverage;  
 this.nhovgOperatingHourAtNoLoadBurningGas = nhovgOperatingHourAtNoLoadBurningGas;  
 this.nhovdOperatingHourAtNoLoadBurningOil = nhovdOperatingHourAtNoLoadBurningOil;  
 this.gt1OperatingHour = gt1OperatingHour;  
 this.gt2OperatingHour = gt2OperatingHour;  
 this.netOutputCorrectedToSummerDesignCfe = netOutputCorrectedToSummerDesignCfe;  
 this.cnGas = cnGas;  
 this.cnOil = cnOil;  
 this.spare2 = spare2;  
 this.gt152gOn = gt152gOn;  
 this.gt252gOn = gt252gOn;  
 this.EA1010400kvSSCbGt1Abn = EA1010400kvSSCbGt1Abn;  
 this.EA1010400kvSSCbGt1Close = EA1010400kvSSCbGt1Close;  
 this.EA1010400kvSSCbGt1Open = EA1010400kvSSCbGt1Open;  
 this.EA1020400kvSSCbGt2Abn = EA1020400kvSSCbGt2Abn;  
 this.EA1020400kvSSCbGt2Close = EA1020400kvSSCbGt2Close;  
 this.EA1020400kvSSCbGt2Open = EA1020400kvSSCbGt2Open;  
 this.EA1030400kvSSCbStClose = EA1030400kvSSCbStClose;  
 this.EA1030400kvSSCbStOpen = EA1030400kvSSCbStOpen;  
 this.gt1Synchronize = gt1Synchronize;  
 this.gt2Synchronize = gt2Synchronize;  
 this.allTariffMeterAbnormal = allTariffMeterAbnormal;  
 this.gt1TransmitterWatAndVarAbnormal = gt1TransmitterWatAndVarAbnormal;  
 this.gt2TransmitterWatAndVarAbnormal = gt2TransmitterWatAndVarAbnormal;  
 this.stTransmitterWatAndVarAbnormal = stTransmitterWatAndVarAbnormal;  
 this.no1TempTransmitterAbnormal = no1TempTransmitterAbnormal;  
 this.no2TempTransmitterAbnormal = no2TempTransmitterAbnormal;  
 this.tempTransmitterSelect0No21No1 = tempTransmitterSelect0No21No1;  
 this.no1HumidityTransmitterAbnormal = no1HumidityTransmitterAbnormal;  
 this.no2HumidityTransmitterAbnormal = no2HumidityTransmitterAbnormal;  
 this.humidityTransmitterSelect0No21No1 = humidityTransmitterSelect0No21No1;  
 this.no1AmbientPressureTransmitterAbnormal = no1AmbientPressureTransmitterAbnormal;  
 this.no2AmbientPressureTransmitterAbnormal = no2AmbientPressureTransmitterAbnormal;  
 this.ambientPressureTransmitterSelect0No21No1 = ambientPressureTransmitterSelect0No21No1;  
 this.no1SeaWaterTemperatureTransmitterAbnormal = no1SeaWaterTemperatureTransmitterAbnormal;  
 this.no2SeaWaterTemperatureTransmitterAbnormal = no2SeaWaterTemperatureTransmitterAbnormal;  
 this.seaWaterTemperatureTransmitterSelect0No21No1 = seaWaterTemperatureTransmitterSelect0No21No1;  
 this.gt1GasSelect = gt1GasSelect;  
 this.gt2GasSelect = gt2GasSelect;  
 this.gt1OilSelect = gt1OilSelect;  
 this.gt2OilSelect = gt2OilSelect;  
 this.no1GtFcbSelect = no1GtFcbSelect;  
 this.no2GtFcbSelect = no2GtFcbSelect;  
 this.fcbCommand = fcbCommand;  
 this.gt1GasComb = gt1GasComb;  
 this.gt2GasComb = gt2GasComb;  
 this.gt1OilComb = gt1OilComb;  
 this.gt2OilComb = gt2OilComb;  
 this.gt1EncenderComb = gt1EncenderComb;  
 this.gt2EncenderComb = gt2EncenderComb;  
 this.actualHigherHeatingValueWet = actualHigherHeatingValueWet;  
 this.actualLowerHeatingValueWet = actualLowerHeatingValueWet;  
 this.actualHigherHeatingValueWet2 = actualHigherHeatingValueWet2;  
 this.actualLowerHeatingValueWet2 = actualLowerHeatingValueWet2;  
 this.gasVolumeTotal = gasVolumeTotal;  
 this.gasMassTotal = gasMassTotal;  
 this.qg55minHeatInputWithGas = qg55minHeatInputWithGas;  
 this.qd55minHeatInputWithOil = qd55minHeatInputWithOil;  
 this.gt1OperationMode = gt1OperationMode;  
 this.gt2OperationMode = gt2OperationMode;  
 this.line1WattHourMas5minInUse = line1WattHourMas5minInUse;  
 this.line2WattHourMas5minInUse = line2WattHourMas5minInUse;  
 this.ctov = ctov;  
 this.line1VarHour5minKvarh = line1VarHour5minKvarh;  
 this.line2VarHour5minKvarh = line2VarHour5minKvarh;  
 this.line1VarHour5minKvarhBackup = line1VarHour5minKvarhBackup;  
 this.line2VarHour5minKvarhBackup = line2VarHour5minKvarhBackup;  
 this.cn = cn;  
 this.cng100corrCng = cng100corrCng;  
 this.line1WattHourMenos5minInUse = line1WattHourMenos5minInUse;  
 this.line2WattHourMenos5minInUse = line2WattHourMenos5minInUse;  
 this.kilovarhourPrevious = kilovarhourPrevious;  
 this.line1KilovarhourPreviousProgq1InUse = line1KilovarhourPreviousProgq1InUse;  
 this.line1KilovarhourPreviousProgq4InUse = line1KilovarhourPreviousProgq4InUse;  
 this.line2KilovarhourPreviousProgq1InUse = line2KilovarhourPreviousProgq1InUse;  
 this.line2KilovarhourPreviousProgq4InUse = line2KilovarhourPreviousProgq4InUse;  
 this.line1KilovarhourPreviousProgq1Main = line1KilovarhourPreviousProgq1Main;  
 this.line1KilovarhourPreviousProgq4Main = line1KilovarhourPreviousProgq4Main;  
 this.line2KilovarhourPreviousProgq1Main = line2KilovarhourPreviousProgq1Main;  
 this.line2KilovarhourPreviousProgq4Main = line2KilovarhourPreviousProgq4Main;  
 this.line1KilovarhourPreviousProgq1Backup = line1KilovarhourPreviousProgq1Backup;  
 this.line1KilovarhourPreviousProgq4Backup = line1KilovarhourPreviousProgq4Backup;  
 this.line2KilovarhourPreviousProgq1Backup = line2KilovarhourPreviousProgq1Backup;  
 this.line2KilovarhourPreviousProgq4Backup = line2KilovarhourPreviousProgq4Backup;  
 this.tipoCarga = tipoCarga;  
 this.cargaId = cargaId;  
 this.horaCarga = horaCarga;  
 this.estatusLimpio = estatusLimpio;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="CRUDOS\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getCrudosId() {  
 return this.crudosId;  
 }  
   
 public void setCrudosId(BigDecimal crudosId) {  
 this.crudosId = crudosId;  
 }  
  
   
 @Column(name="DIA", length=200)  
 public String getDia() {  
 return this.dia;  
 }  
   
 public void setDia(String dia) {  
 this.dia = dia;  
 }  
  
   
 @Column(name="MINUTO", length=20)  
 public String getMinuto() {  
 return this.minuto;  
 }  
   
 public void setMinuto(String minuto) {  
 this.minuto = minuto;  
 }  
  
   
 @Column(name="FUEL\_SELECT", length=100)  
 public String getFuelSelect() {  
 return this.fuelSelect;  
 }  
   
 public void setFuelSelect(String fuelSelect) {  
 this.fuelSelect = fuelSelect;  
 }  
  
   
 @Column(name="HOURLY\_LOAD\_IN\_USE", length=100)  
 public String getHourlyLoadInUse() {  
 return this.hourlyLoadInUse;  
 }  
   
 public void setHourlyLoadInUse(String hourlyLoadInUse) {  
 this.hourlyLoadInUse = hourlyLoadInUse;  
 }  
  
   
 @Column(name="LINE1\_WATT\_HOUR\_MAS\_5MIN", length=100)  
 public String getLine1WattHourMas5min() {  
 return this.line1WattHourMas5min;  
 }  
   
 public void setLine1WattHourMas5min(String line1WattHourMas5min) {  
 this.line1WattHourMas5min = line1WattHourMas5min;  
 }  
  
   
 @Column(name="LINE2\_WATT\_HOUR\_MAS\_5MIN", length=100)  
 public String getLine2WattHourMas5min() {  
 return this.line2WattHourMas5min;  
 }  
   
 public void setLine2WattHourMas5min(String line2WattHourMas5min) {  
 this.line2WattHourMas5min = line2WattHourMas5min;  
 }  
  
   
 @Column(name="LINE1\_WATT\_HOUR\_MENOS\_5MIN", length=100)  
 public String getLine1WattHourMenos5min() {  
 return this.line1WattHourMenos5min;  
 }  
   
 public void setLine1WattHourMenos5min(String line1WattHourMenos5min) {  
 this.line1WattHourMenos5min = line1WattHourMenos5min;  
 }  
  
   
 @Column(name="LINE2\_WATT\_HOUR\_MENOS\_5MIN", length=100)  
 public String getLine2WattHourMenos5min() {  
 return this.line2WattHourMenos5min;  
 }  
   
 public void setLine2WattHourMenos5min(String line2WattHourMenos5min) {  
 this.line2WattHourMenos5min = line2WattHourMenos5min;  
 }  
  
   
 @Column(name="LINE1\_WATT\_HOUR\_MAS\_5MIN\_BACKUP", length=100)  
 public String getLine1WattHourMas5minBackup() {  
 return this.line1WattHourMas5minBackup;  
 }  
   
 public void setLine1WattHourMas5minBackup(String line1WattHourMas5minBackup) {  
 this.line1WattHourMas5minBackup = line1WattHourMas5minBackup;  
 }  
  
   
 @Column(name="LINE2\_WATT\_HOUR\_MAS\_5MIN\_BACKUP", length=100)  
 public String getLine2WattHourMas5minBackup() {  
 return this.line2WattHourMas5minBackup;  
 }  
   
 public void setLine2WattHourMas5minBackup(String line2WattHourMas5minBackup) {  
 this.line2WattHourMas5minBackup = line2WattHourMas5minBackup;  
 }  
  
   
 @Column(name="LINE1\_WATT\_HOUR\_MENOS\_5MIN\_BACKUP", length=100)  
 public String getLine1WattHourMenos5minBackup() {  
 return this.line1WattHourMenos5minBackup;  
 }  
   
 public void setLine1WattHourMenos5minBackup(String line1WattHourMenos5minBackup) {  
 this.line1WattHourMenos5minBackup = line1WattHourMenos5minBackup;  
 }  
  
   
 @Column(name="LINE2\_WATT\_HOUR\_MENOS\_5MIN\_BACKUP", length=100)  
 public String getLine2WattHourMenos5minBackup() {  
 return this.line2WattHourMenos5minBackup;  
 }  
   
 public void setLine2WattHourMenos5minBackup(String line2WattHourMenos5minBackup) {  
 this.line2WattHourMenos5minBackup = line2WattHourMenos5minBackup;  
 }  
  
   
 @Column(name="GT1\_ACTIVE\_POWER", length=100)  
 public String getGt1ActivePower() {  
 return this.gt1ActivePower;  
 }  
   
 public void setGt1ActivePower(String gt1ActivePower) {  
 this.gt1ActivePower = gt1ActivePower;  
 }  
  
   
 @Column(name="GT2\_ACTIVE\_POWER", length=100)  
 public String getGt2ActivePower() {  
 return this.gt2ActivePower;  
 }  
   
 public void setGt2ActivePower(String gt2ActivePower) {  
 this.gt2ActivePower = gt2ActivePower;  
 }  
  
   
 @Column(name="ST\_ACTIVE\_POWER", length=100)  
 public String getStActivePower() {  
 return this.stActivePower;  
 }  
   
 public void setStActivePower(String stActivePower) {  
 this.stActivePower = stActivePower;  
 }  
  
   
 @Column(name="GT1\_REACTIVE\_POWER", length=100)  
 public String getGt1ReactivePower() {  
 return this.gt1ReactivePower;  
 }  
   
 public void setGt1ReactivePower(String gt1ReactivePower) {  
 this.gt1ReactivePower = gt1ReactivePower;  
 }  
  
   
 @Column(name="GT2\_REACTIVE\_POWER", length=100)  
 public String getGt2ReactivePower() {  
 return this.gt2ReactivePower;  
 }  
   
 public void setGt2ReactivePower(String gt2ReactivePower) {  
 this.gt2ReactivePower = gt2ReactivePower;  
 }  
  
   
 @Column(name="ST\_REACTIVE\_POWER", length=100)  
 public String getStReactivePower() {  
 return this.stReactivePower;  
 }  
   
 public void setStReactivePower(String stReactivePower) {  
 this.stReactivePower = stReactivePower;  
 }  
  
   
 @Column(name="LINE1\_WATT\_INSTANT", length=100)  
 public String getLine1WattInstant() {  
 return this.line1WattInstant;  
 }  
   
 public void setLine1WattInstant(String line1WattInstant) {  
 this.line1WattInstant = line1WattInstant;  
 }  
  
   
 @Column(name="LINE2\_WATT\_INSTANT", length=100)  
 public String getLine2WattInstant() {  
 return this.line2WattInstant;  
 }  
   
 public void setLine2WattInstant(String line2WattInstant) {  
 this.line2WattInstant = line2WattInstant;  
 }  
  
   
 @Column(name="LINE1\_VAR\_INSTANT", length=100)  
 public String getLine1VarInstant() {  
 return this.line1VarInstant;  
 }  
   
 public void setLine1VarInstant(String line1VarInstant) {  
 this.line1VarInstant = line1VarInstant;  
 }  
  
   
 @Column(name="LINE2\_VAR\_INSTANT", length=100)  
 public String getLine2VarInstant() {  
 return this.line2VarInstant;  
 }  
   
 public void setLine2VarInstant(String line2VarInstant) {  
 this.line2VarInstant = line2VarInstant;  
 }  
  
   
 @Column(name="PLANT\_NET\_WATT\_HOUR\_CALCULATION\_FROM\_GROSS", length=100)  
 public String getPlantNetWattHourCalculationFromGross() {  
 return this.plantNetWattHourCalculationFromGross;  
 }  
   
 public void setPlantNetWattHourCalculationFromGross(String plantNetWattHourCalculationFromGross) {  
 this.plantNetWattHourCalculationFromGross = plantNetWattHourCalculationFromGross;  
 }  
  
   
 @Column(name="STG\_WATT\_HOUR", length=100)  
 public String getStgWattHour() {  
 return this.stgWattHour;  
 }  
   
 public void setStgWattHour(String stgWattHour) {  
 this.stgWattHour = stgWattHour;  
 }  
  
   
 @Column(name="STG\_VAR\_HOUR", length=100)  
 public String getStgVarHour() {  
 return this.stgVarHour;  
 }  
   
 public void setStgVarHour(String stgVarHour) {  
 this.stgVarHour = stgVarHour;  
 }  
  
   
 @Column(name="NO1\_GTG\_WATT\_HOUR", length=100)  
 public String getNo1GtgWattHour() {  
 return this.no1GtgWattHour;  
 }  
   
 public void setNo1GtgWattHour(String no1GtgWattHour) {  
 this.no1GtgWattHour = no1GtgWattHour;  
 }  
  
   
 @Column(name="NO1\_GTG\_VAR\_HOUR", length=100)  
 public String getNo1GtgVarHour() {  
 return this.no1GtgVarHour;  
 }  
   
 public void setNo1GtgVarHour(String no1GtgVarHour) {  
 this.no1GtgVarHour = no1GtgVarHour;  
 }  
  
   
 @Column(name="NO1\_GTG\_UNIT\_HOUSE\_LOAD", length=100)  
 public String getNo1GtgUnitHouseLoad() {  
 return this.no1GtgUnitHouseLoad;  
 }  
   
 public void setNo1GtgUnitHouseLoad(String no1GtgUnitHouseLoad) {  
 this.no1GtgUnitHouseLoad = no1GtgUnitHouseLoad;  
 }  
  
   
 @Column(name="NO2\_GTG\_WATT\_HOUR", length=100)  
 public String getNo2GtgWattHour() {  
 return this.no2GtgWattHour;  
 }  
   
 public void setNo2GtgWattHour(String no2GtgWattHour) {  
 this.no2GtgWattHour = no2GtgWattHour;  
 }  
  
   
 @Column(name="NO2\_GTG\_VAR\_HOUR", length=100)  
 public String getNo2GtgVarHour() {  
 return this.no2GtgVarHour;  
 }  
   
 public void setNo2GtgVarHour(String no2GtgVarHour) {  
 this.no2GtgVarHour = no2GtgVarHour;  
 }  
  
   
 @Column(name="NO2\_GTG\_UNIT\_HOUSE\_LOAD", length=100)  
 public String getNo2GtgUnitHouseLoad() {  
 return this.no2GtgUnitHouseLoad;  
 }  
   
 public void setNo2GtgUnitHouseLoad(String no2GtgUnitHouseLoad) {  
 this.no2GtgUnitHouseLoad = no2GtgUnitHouseLoad;  
 }  
  
   
 @Column(name="AMBIENT\_TEMPERATURE\_IN\_USE", length=100)  
 public String getAmbientTemperatureInUse() {  
 return this.ambientTemperatureInUse;  
 }  
   
 public void setAmbientTemperatureInUse(String ambientTemperatureInUse) {  
 this.ambientTemperatureInUse = ambientTemperatureInUse;  
 }  
  
   
 @Column(name="AMBIENT\_TEMPERATURE1", length=100)  
 public String getAmbientTemperature1() {  
 return this.ambientTemperature1;  
 }  
   
 public void setAmbientTemperature1(String ambientTemperature1) {  
 this.ambientTemperature1 = ambientTemperature1;  
 }  
  
   
 @Column(name="AMBIENT\_TEMPERATURE2", length=100)  
 public String getAmbientTemperature2() {  
 return this.ambientTemperature2;  
 }  
   
 public void setAmbientTemperature2(String ambientTemperature2) {  
 this.ambientTemperature2 = ambientTemperature2;  
 }  
  
   
 @Column(name="RELATIVE\_HUMIDITY\_IN\_USE", length=100)  
 public String getRelativeHumidityInUse() {  
 return this.relativeHumidityInUse;  
 }  
   
 public void setRelativeHumidityInUse(String relativeHumidityInUse) {  
 this.relativeHumidityInUse = relativeHumidityInUse;  
 }  
  
   
 @Column(name="RELATIVE\_HUMIDITY1", length=100)  
 public String getRelativeHumidity1() {  
 return this.relativeHumidity1;  
 }  
   
 public void setRelativeHumidity1(String relativeHumidity1) {  
 this.relativeHumidity1 = relativeHumidity1;  
 }  
  
   
 @Column(name="RELATIVE\_HUMIDITY2", length=100)  
 public String getRelativeHumidity2() {  
 return this.relativeHumidity2;  
 }  
   
 public void setRelativeHumidity2(String relativeHumidity2) {  
 this.relativeHumidity2 = relativeHumidity2;  
 }  
  
   
 @Column(name="AMBIENT\_PRESSURE\_IN\_USE", length=100)  
 public String getAmbientPressureInUse() {  
 return this.ambientPressureInUse;  
 }  
   
 public void setAmbientPressureInUse(String ambientPressureInUse) {  
 this.ambientPressureInUse = ambientPressureInUse;  
 }  
  
   
 @Column(name="AMBIENT\_PRESSURE1", length=100)  
 public String getAmbientPressure1() {  
 return this.ambientPressure1;  
 }  
   
 public void setAmbientPressure1(String ambientPressure1) {  
 this.ambientPressure1 = ambientPressure1;  
 }  
  
   
 @Column(name="AMBIENT\_PRESSURE2", length=100)  
 public String getAmbientPressure2() {  
 return this.ambientPressure2;  
 }  
   
 public void setAmbientPressure2(String ambientPressure2) {  
 this.ambientPressure2 = ambientPressure2;  
 }  
  
   
 @Column(name="SEA\_WATER\_TEMPERATURE\_IN\_USE", length=100)  
 public String getSeaWaterTemperatureInUse() {  
 return this.seaWaterTemperatureInUse;  
 }  
   
 public void setSeaWaterTemperatureInUse(String seaWaterTemperatureInUse) {  
 this.seaWaterTemperatureInUse = seaWaterTemperatureInUse;  
 }  
  
   
 @Column(name="SEA\_WATER\_TEMPERATURE1", length=100)  
 public String getSeaWaterTemperature1() {  
 return this.seaWaterTemperature1;  
 }  
   
 public void setSeaWaterTemperature1(String seaWaterTemperature1) {  
 this.seaWaterTemperature1 = seaWaterTemperature1;  
 }  
  
   
 @Column(name="SEA\_WATER\_TEMPERATURE2", length=100)  
 public String getSeaWaterTemperature2() {  
 return this.seaWaterTemperature2;  
 }  
   
 public void setSeaWaterTemperature2(String seaWaterTemperature2) {  
 this.seaWaterTemperature2 = seaWaterTemperature2;  
 }  
  
   
 @Column(name="SPARE", length=100)  
 public String getSpare() {  
 return this.spare;  
 }  
   
 public void setSpare(String spare) {  
 this.spare = spare;  
 }  
  
   
 @Column(name="FUEL\_GAS\_HIGHER\_HEATING\_VALUE", length=100)  
 public String getFuelGasHigherHeatingValue() {  
 return this.fuelGasHigherHeatingValue;  
 }  
   
 public void setFuelGasHigherHeatingValue(String fuelGasHigherHeatingValue) {  
 this.fuelGasHigherHeatingValue = fuelGasHigherHeatingValue;  
 }  
  
   
 @Column(name="FUEL\_GAS\_LOWER\_HEATING\_VALUE", length=100)  
 public String getFuelGasLowerHeatingValue() {  
 return this.fuelGasLowerHeatingValue;  
 }  
   
 public void setFuelGasLowerHeatingValue(String fuelGasLowerHeatingValue) {  
 this.fuelGasLowerHeatingValue = fuelGasLowerHeatingValue;  
 }  
  
   
 @Column(name="FUEL\_GAS\_HIGHER\_HEATING\_VALUE2", length=100)  
 public String getFuelGasHigherHeatingValue2() {  
 return this.fuelGasHigherHeatingValue2;  
 }  
   
 public void setFuelGasHigherHeatingValue2(String fuelGasHigherHeatingValue2) {  
 this.fuelGasHigherHeatingValue2 = fuelGasHigherHeatingValue2;  
 }  
  
   
 @Column(name="FUEL\_GAS\_LOWER\_HEATING\_VALUE2", length=100)  
 public String getFuelGasLowerHeatingValue2() {  
 return this.fuelGasLowerHeatingValue2;  
 }  
   
 public void setFuelGasLowerHeatingValue2(String fuelGasLowerHeatingValue2) {  
 this.fuelGasLowerHeatingValue2 = fuelGasLowerHeatingValue2;  
 }  
  
   
 @Column(name="FUEL\_OIL\_HIGHER\_HEATING\_VALEU", length=100)  
 public String getFuelOilHigherHeatingValeu() {  
 return this.fuelOilHigherHeatingValeu;  
 }  
   
 public void setFuelOilHigherHeatingValeu(String fuelOilHigherHeatingValeu) {  
 this.fuelOilHigherHeatingValeu = fuelOilHigherHeatingValeu;  
 }  
  
   
 @Column(name="FUEL\_OIL\_LOWER\_HEATING\_VALUE", length=100)  
 public String getFuelOilLowerHeatingValue() {  
 return this.fuelOilLowerHeatingValue;  
 }  
   
 public void setFuelOilLowerHeatingValue(String fuelOilLowerHeatingValue) {  
 this.fuelOilLowerHeatingValue = fuelOilLowerHeatingValue;  
 }  
  
   
 @Column(name="FUEL\_OIL\_HIGHER\_HEATING\_VALEU2", length=100)  
 public String getFuelOilHigherHeatingValeu2() {  
 return this.fuelOilHigherHeatingValeu2;  
 }  
   
 public void setFuelOilHigherHeatingValeu2(String fuelOilHigherHeatingValeu2) {  
 this.fuelOilHigherHeatingValeu2 = fuelOilHigherHeatingValeu2;  
 }  
  
   
 @Column(name="FUEL\_OIL\_LOWER\_HEATING\_VALUE2", length=100)  
 public String getFuelOilLowerHeatingValue2() {  
 return this.fuelOilLowerHeatingValue2;  
 }  
   
 public void setFuelOilLowerHeatingValue2(String fuelOilLowerHeatingValue2) {  
 this.fuelOilLowerHeatingValue2 = fuelOilLowerHeatingValue2;  
 }  
  
   
 @Column(name="FUEL\_OIL\_SPECIFIC\_GRAVITY", length=100)  
 public String getFuelOilSpecificGravity() {  
 return this.fuelOilSpecificGravity;  
 }  
   
 public void setFuelOilSpecificGravity(String fuelOilSpecificGravity) {  
 this.fuelOilSpecificGravity = fuelOilSpecificGravity;  
 }  
  
   
 @Column(name="FUEL\_GAS\_FLOW\_CURRENT\_MAS\_FLOW", length=100)  
 public String getFuelGasFlowCurrentMasFlow() {  
 return this.fuelGasFlowCurrentMasFlow;  
 }  
   
 public void setFuelGasFlowCurrentMasFlow(String fuelGasFlowCurrentMasFlow) {  
 this.fuelGasFlowCurrentMasFlow = fuelGasFlowCurrentMasFlow;  
 }  
  
   
 @Column(name="STRM1\_CURRENT\_VOLUME\_FLOW\_RATE", length=100)  
 public String getStrm1CurrentVolumeFlowRate() {  
 return this.strm1CurrentVolumeFlowRate;  
 }  
   
 public void setStrm1CurrentVolumeFlowRate(String strm1CurrentVolumeFlowRate) {  
 this.strm1CurrentVolumeFlowRate = strm1CurrentVolumeFlowRate;  
 }  
  
   
 @Column(name="FUEL\_GAS\_FLOW\_5MIN\_ACCUMULATED\_VALUE", length=100)  
 public String getFuelGasFlow5minAccumulatedValue() {  
 return this.fuelGasFlow5minAccumulatedValue;  
 }  
   
 public void setFuelGasFlow5minAccumulatedValue(String fuelGasFlow5minAccumulatedValue) {  
 this.fuelGasFlow5minAccumulatedValue = fuelGasFlow5minAccumulatedValue;  
 }  
  
   
 @Column(name="OIL\_FLOW\_GT1\_GT2", length=100)  
 public String getOilFlowGt1Gt2() {  
 return this.oilFlowGt1Gt2;  
 }  
   
 public void setOilFlowGt1Gt2(String oilFlowGt1Gt2) {  
 this.oilFlowGt1Gt2 = oilFlowGt1Gt2;  
 }  
  
   
 @Column(name="GT1\_OIL\_FLOW\_CURRENT\_VALUE", length=100)  
 public String getGt1OilFlowCurrentValue() {  
 return this.gt1OilFlowCurrentValue;  
 }  
   
 public void setGt1OilFlowCurrentValue(String gt1OilFlowCurrentValue) {  
 this.gt1OilFlowCurrentValue = gt1OilFlowCurrentValue;  
 }  
  
   
 @Column(name="GT2\_OIL\_FLOW\_CURRENT\_VALUE", length=100)  
 public String getGt2OilFlowCurrentValue() {  
 return this.gt2OilFlowCurrentValue;  
 }  
   
 public void setGt2OilFlowCurrentValue(String gt2OilFlowCurrentValue) {  
 this.gt2OilFlowCurrentValue = gt2OilFlowCurrentValue;  
 }  
  
   
 @Column(name="OIL\_FLOW\_GT1\_GT22", length=100)  
 public String getOilFlowGt1Gt22() {  
 return this.oilFlowGt1Gt22;  
 }  
   
 public void setOilFlowGt1Gt22(String oilFlowGt1Gt22) {  
 this.oilFlowGt1Gt22 = oilFlowGt1Gt22;  
 }  
  
   
 @Column(name="GT1\_OIL\_FLOW\_5MIN\_ACCUMULATED\_VALUE", length=100)  
 public String getGt1OilFlow5minAccumulatedValue() {  
 return this.gt1OilFlow5minAccumulatedValue;  
 }  
   
 public void setGt1OilFlow5minAccumulatedValue(String gt1OilFlow5minAccumulatedValue) {  
 this.gt1OilFlow5minAccumulatedValue = gt1OilFlow5minAccumulatedValue;  
 }  
  
   
 @Column(name="GT2\_OIL\_FLOW\_5MIN\_ACCUMULATED\_VALUE", length=100)  
 public String getGt2OilFlow5minAccumulatedValue() {  
 return this.gt2OilFlow5minAccumulatedValue;  
 }  
   
 public void setGt2OilFlow5minAccumulatedValue(String gt2OilFlow5minAccumulatedValue) {  
 this.gt2OilFlow5minAccumulatedValue = gt2OilFlow5minAccumulatedValue;  
 }  
  
   
 @Column(name="OVERALL\_POWER\_FACTOR", length=100)  
 public String getOverallPowerFactor() {  
 return this.overallPowerFactor;  
 }  
   
 public void setOverallPowerFactor(String overallPowerFactor) {  
 this.overallPowerFactor = overallPowerFactor;  
 }  
  
   
 @Column(name="NET\_POWER\_OF\_5MIN\_5MIN\_ACCUMULATED\_WITH\_GAS\_FIRING", length=100)  
 public String getNetPowerOf5min5minAccumulatedWithGasFiring() {  
 return this.netPowerOf5min5minAccumulatedWithGasFiring;  
 }  
   
 public void setNetPowerOf5min5minAccumulatedWithGasFiring(String netPowerOf5min5minAccumulatedWithGasFiring) {  
 this.netPowerOf5min5minAccumulatedWithGasFiring = netPowerOf5min5minAccumulatedWithGasFiring;  
 }  
  
   
 @Column(name="NET\_POWER\_OF\_5MIN\_5MIN\_ACCUMULATED\_WITH\_OIL\_FIRING", length=100)  
 public String getNetPowerOf5min5minAccumulatedWithOilFiring() {  
 return this.netPowerOf5min5minAccumulatedWithOilFiring;  
 }  
   
 public void setNetPowerOf5min5minAccumulatedWithOilFiring(String netPowerOf5min5minAccumulatedWithOilFiring) {  
 this.netPowerOf5min5minAccumulatedWithOilFiring = netPowerOf5min5minAccumulatedWithOilFiring;  
 }  
  
   
 @Column(name="LOAD\_PERCENTAGE\_AT\_GAS", length=100)  
 public String getLoadPercentageAtGas() {  
 return this.loadPercentageAtGas;  
 }  
   
 public void setLoadPercentageAtGas(String loadPercentageAtGas) {  
 this.loadPercentageAtGas = loadPercentageAtGas;  
 }  
  
   
 @Column(name="LOAD\_PERCENTAGE\_AT\_OIL", length=100)  
 public String getLoadPercentageAtOil() {  
 return this.loadPercentageAtOil;  
 }  
   
 public void setLoadPercentageAtOil(String loadPercentageAtOil) {  
 this.loadPercentageAtOil = loadPercentageAtOil;  
 }  
  
   
 @Column(name="CTUNG\_AT\_YEAR\_AVERAGE\_DESIGN\_CONDITION\_GAS", length=100)  
 public String getCtungAtYearAverageDesignConditionGas() {  
 return this.ctungAtYearAverageDesignConditionGas;  
 }  
   
 public void setCtungAtYearAverageDesignConditionGas(String ctungAtYearAverageDesignConditionGas) {  
 this.ctungAtYearAverageDesignConditionGas = ctungAtYearAverageDesignConditionGas;  
 }  
  
   
 @Column(name="CTUNG\_AT\_YEAR\_AVERAGE\_DESIGN\_CONDITION\_OIL", length=100)  
 public String getCtungAtYearAverageDesignConditionOil() {  
 return this.ctungAtYearAverageDesignConditionOil;  
 }  
   
 public void setCtungAtYearAverageDesignConditionOil(String ctungAtYearAverageDesignConditionOil) {  
 this.ctungAtYearAverageDesignConditionOil = ctungAtYearAverageDesignConditionOil;  
 }  
  
   
 @Column(name="CORRECTED\_CTUNG\_GAS", length=100)  
 public String getCorrectedCtungGas() {  
 return this.correctedCtungGas;  
 }  
   
 public void setCorrectedCtungGas(String correctedCtungGas) {  
 this.correctedCtungGas = correctedCtungGas;  
 }  
  
   
 @Column(name="CORRECTED\_CTUNG\_OIL", length=100)  
 public String getCorrectedCtungOil() {  
 return this.correctedCtungOil;  
 }  
   
 public void setCorrectedCtungOil(String correctedCtungOil) {  
 this.correctedCtungOil = correctedCtungOil;  
 }  
  
   
 @Column(name="FUEL\_GAS\_PRESSURE", length=100)  
 public String getFuelGasPressure() {  
 return this.fuelGasPressure;  
 }  
   
 public void setFuelGasPressure(String fuelGasPressure) {  
 this.fuelGasPressure = fuelGasPressure;  
 }  
  
   
 @Column(name="GAS\_TEMPERATURE", length=100)  
 public String getGasTemperature() {  
 return this.gasTemperature;  
 }  
   
 public void setGasTemperature(String gasTemperature) {  
 this.gasTemperature = gasTemperature;  
 }  
  
   
 @Column(name="LOAD\_DEMAND", length=100)  
 public String getLoadDemand() {  
 return this.loadDemand;  
 }  
   
 public void setLoadDemand(String loadDemand) {  
 this.loadDemand = loadDemand;  
 }  
  
   
 @Column(name="EH5\_12\_ROLLING\_AVERAGE", length=100)  
 public String getEh512RollingAverage() {  
 return this.eh512RollingAverage;  
 }  
   
 public void setEh512RollingAverage(String eh512RollingAverage) {  
 this.eh512RollingAverage = eh512RollingAverage;  
 }  
  
   
 @Column(name="AMBIENT\_TEMPERATURE\_12\_ROLLING\_AVERAGE", length=100)  
 public String getAmbientTemperature12RollingAverage() {  
 return this.ambientTemperature12RollingAverage;  
 }  
   
 public void setAmbientTemperature12RollingAverage(String ambientTemperature12RollingAverage) {  
 this.ambientTemperature12RollingAverage = ambientTemperature12RollingAverage;  
 }  
  
   
 @Column(name="RELATIVE\_HUMIDITY\_12\_ROLLING\_AVERAGE", length=100)  
 public String getRelativeHumidity12RollingAverage() {  
 return this.relativeHumidity12RollingAverage;  
 }  
   
 public void setRelativeHumidity12RollingAverage(String relativeHumidity12RollingAverage) {  
 this.relativeHumidity12RollingAverage = relativeHumidity12RollingAverage;  
 }  
  
   
 @Column(name="AMBIENT\_PRESSURE\_12\_ROLLING\_AVERAGE", length=100)  
 public String getAmbientPressure12RollingAverage() {  
 return this.ambientPressure12RollingAverage;  
 }  
   
 public void setAmbientPressure12RollingAverage(String ambientPressure12RollingAverage) {  
 this.ambientPressure12RollingAverage = ambientPressure12RollingAverage;  
 }  
  
   
 @Column(name="CONDENSER\_VACUUM\_12\_ROLLING\_AVERAGE", length=100)  
 public String getCondenserVacuum12RollingAverage() {  
 return this.condenserVacuum12RollingAverage;  
 }  
   
 public void setCondenserVacuum12RollingAverage(String condenserVacuum12RollingAverage) {  
 this.condenserVacuum12RollingAverage = condenserVacuum12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_GAS\_FLOW\_12\_ROLLING\_AVERAGE", length=100)  
 public String getFuelGasFlow12RollingAverage() {  
 return this.fuelGasFlow12RollingAverage;  
 }  
   
 public void setFuelGasFlow12RollingAverage(String fuelGasFlow12RollingAverage) {  
 this.fuelGasFlow12RollingAverage = fuelGasFlow12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_GAS\_FLOW\_12\_ROLLING\_AVERAGE2", length=100)  
 public String getFuelGasFlow12RollingAverage2() {  
 return this.fuelGasFlow12RollingAverage2;  
 }  
   
 public void setFuelGasFlow12RollingAverage2(String fuelGasFlow12RollingAverage2) {  
 this.fuelGasFlow12RollingAverage2 = fuelGasFlow12RollingAverage2;  
 }  
  
   
 @Column(name="FUEL\_GAS\_PRESSURE\_12\_ROLLING\_AVERAGE", length=100)  
 public String getFuelGasPressure12RollingAverage() {  
 return this.fuelGasPressure12RollingAverage;  
 }  
   
 public void setFuelGasPressure12RollingAverage(String fuelGasPressure12RollingAverage) {  
 this.fuelGasPressure12RollingAverage = fuelGasPressure12RollingAverage;  
 }  
  
   
 @Column(name="OVERALL\_POWER\_FACTOR\_12\_ROLLING\_AVERAGE", length=100)  
 public String getOverallPowerFactor12RollingAverage() {  
 return this.overallPowerFactor12RollingAverage;  
 }  
   
 public void setOverallPowerFactor12RollingAverage(String overallPowerFactor12RollingAverage) {  
 this.overallPowerFactor12RollingAverage = overallPowerFactor12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_OIL\_FLOW\_12\_ROLLING\_AVERAGE", length=100)  
 public String getFuelOilFlow12RollingAverage() {  
 return this.fuelOilFlow12RollingAverage;  
 }  
   
 public void setFuelOilFlow12RollingAverage(String fuelOilFlow12RollingAverage) {  
 this.fuelOilFlow12RollingAverage = fuelOilFlow12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_OIL\_FLOW\_12\_ROLLING\_AVERAGE2", length=100)  
 public String getFuelOilFlow12RollingAverage2() {  
 return this.fuelOilFlow12RollingAverage2;  
 }  
   
 public void setFuelOilFlow12RollingAverage2(String fuelOilFlow12RollingAverage2) {  
 this.fuelOilFlow12RollingAverage2 = fuelOilFlow12RollingAverage2;  
 }  
  
   
 @Column(name="CORRECTED\_CTUNG\_12\_ROLLING\_WEIGHTED\_AVERAGE\_GAS", length=100)  
 public String getCorrectedCtung12RollingWeightedAverageGas() {  
 return this.correctedCtung12RollingWeightedAverageGas;  
 }  
   
 public void setCorrectedCtung12RollingWeightedAverageGas(String correctedCtung12RollingWeightedAverageGas) {  
 this.correctedCtung12RollingWeightedAverageGas = correctedCtung12RollingWeightedAverageGas;  
 }  
  
   
 @Column(name="CORRECTED\_CTUNG\_12\_ROLLING\_WEIGHTED\_AVERAGE\_OIL", length=100)  
 public String getCorrectedCtung12RollingWeightedAverageOil() {  
 return this.correctedCtung12RollingWeightedAverageOil;  
 }  
   
 public void setCorrectedCtung12RollingWeightedAverageOil(String correctedCtung12RollingWeightedAverageOil) {  
 this.correctedCtung12RollingWeightedAverageOil = correctedCtung12RollingWeightedAverageOil;  
 }  
  
   
 @Column(name="FUEL\_GAS\_HIGHER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE", length=100)  
 public String getFuelGasHigherHeatingValue12RollingAverage() {  
 return this.fuelGasHigherHeatingValue12RollingAverage;  
 }  
   
 public void setFuelGasHigherHeatingValue12RollingAverage(String fuelGasHigherHeatingValue12RollingAverage) {  
 this.fuelGasHigherHeatingValue12RollingAverage = fuelGasHigherHeatingValue12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_GAS\_LOWER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE", length=100)  
 public String getFuelGasLowerHeatingValue12RollingAverage() {  
 return this.fuelGasLowerHeatingValue12RollingAverage;  
 }  
   
 public void setFuelGasLowerHeatingValue12RollingAverage(String fuelGasLowerHeatingValue12RollingAverage) {  
 this.fuelGasLowerHeatingValue12RollingAverage = fuelGasLowerHeatingValue12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_GAS\_HIGHER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE2", length=100)  
 public String getFuelGasHigherHeatingValue12RollingAverage2() {  
 return this.fuelGasHigherHeatingValue12RollingAverage2;  
 }  
   
 public void setFuelGasHigherHeatingValue12RollingAverage2(String fuelGasHigherHeatingValue12RollingAverage2) {  
 this.fuelGasHigherHeatingValue12RollingAverage2 = fuelGasHigherHeatingValue12RollingAverage2;  
 }  
  
   
 @Column(name="FUEL\_GAS\_LOWER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE2", length=100)  
 public String getFuelGasLowerHeatingValue12RollingAverage2() {  
 return this.fuelGasLowerHeatingValue12RollingAverage2;  
 }  
   
 public void setFuelGasLowerHeatingValue12RollingAverage2(String fuelGasLowerHeatingValue12RollingAverage2) {  
 this.fuelGasLowerHeatingValue12RollingAverage2 = fuelGasLowerHeatingValue12RollingAverage2;  
 }  
  
   
 @Column(name="FUEL\_OIL\_HIGHER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE", length=100)  
 public String getFuelOilHigherHeatingValue12RollingAverage() {  
 return this.fuelOilHigherHeatingValue12RollingAverage;  
 }  
   
 public void setFuelOilHigherHeatingValue12RollingAverage(String fuelOilHigherHeatingValue12RollingAverage) {  
 this.fuelOilHigherHeatingValue12RollingAverage = fuelOilHigherHeatingValue12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_OIL\_LOWER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE", length=100)  
 public String getFuelOilLowerHeatingValue12RollingAverage() {  
 return this.fuelOilLowerHeatingValue12RollingAverage;  
 }  
   
 public void setFuelOilLowerHeatingValue12RollingAverage(String fuelOilLowerHeatingValue12RollingAverage) {  
 this.fuelOilLowerHeatingValue12RollingAverage = fuelOilLowerHeatingValue12RollingAverage;  
 }  
  
   
 @Column(name="FUEL\_OIL\_HIGHER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE2", length=100)  
 public String getFuelOilHigherHeatingValue12RollingAverage2() {  
 return this.fuelOilHigherHeatingValue12RollingAverage2;  
 }  
   
 public void setFuelOilHigherHeatingValue12RollingAverage2(String fuelOilHigherHeatingValue12RollingAverage2) {  
 this.fuelOilHigherHeatingValue12RollingAverage2 = fuelOilHigherHeatingValue12RollingAverage2;  
 }  
  
   
 @Column(name="FUEL\_OIL\_LOWER\_HEATING\_VALUE\_12\_ROLLING\_AVERAGE2", length=100)  
 public String getFuelOilLowerHeatingValue12RollingAverage2() {  
 return this.fuelOilLowerHeatingValue12RollingAverage2;  
 }  
   
 public void setFuelOilLowerHeatingValue12RollingAverage2(String fuelOilLowerHeatingValue12RollingAverage2) {  
 this.fuelOilLowerHeatingValue12RollingAverage2 = fuelOilLowerHeatingValue12RollingAverage2;  
 }  
  
   
 @Column(name="NUMBER\_OF\_MINUTES\_OF\_OPERATING\_AT\_GAS\_FIRING", length=100)  
 public String getNumberOfMinutesOfOperatingAtGasFiring() {  
 return this.numberOfMinutesOfOperatingAtGasFiring;  
 }  
   
 public void setNumberOfMinutesOfOperatingAtGasFiring(String numberOfMinutesOfOperatingAtGasFiring) {  
 this.numberOfMinutesOfOperatingAtGasFiring = numberOfMinutesOfOperatingAtGasFiring;  
 }  
  
   
 @Column(name="NUMBER\_OF\_MINUTES\_OF\_OPERATING\_AT\_OIL\_FIRING", length=100)  
 public String getNumberOfMinutesOfOperatingAtOilFiring() {  
 return this.numberOfMinutesOfOperatingAtOilFiring;  
 }  
   
 public void setNumberOfMinutesOfOperatingAtOilFiring(String numberOfMinutesOfOperatingAtOilFiring) {  
 this.numberOfMinutesOfOperatingAtOilFiring = numberOfMinutesOfOperatingAtOilFiring;  
 }  
  
   
 @Column(name="EH\_12\_ROLLING\_AVERAGE", length=100)  
 public String getEh12RollingAverage() {  
 return this.eh12RollingAverage;  
 }  
   
 public void setEh12RollingAverage(String eh12RollingAverage) {  
 this.eh12RollingAverage = eh12RollingAverage;  
 }  
  
   
 @Column(name="EGH\_12\_ROLLING\_AVERAGE", length=100)  
 public String getEgh12RollingAverage() {  
 return this.egh12RollingAverage;  
 }  
   
 public void setEgh12RollingAverage(String egh12RollingAverage) {  
 this.egh12RollingAverage = egh12RollingAverage;  
 }  
  
   
 @Column(name="EDH\_12\_ROLLING\_AVERAGE", length=100)  
 public String getEdh12RollingAverage() {  
 return this.edh12RollingAverage;  
 }  
   
 public void setEdh12RollingAverage(String edh12RollingAverage) {  
 this.edh12RollingAverage = edh12RollingAverage;  
 }  
  
   
 @Column(name="NHOVG\_OPERATING\_HOUR\_AT\_NO\_LOAD\_BURNING\_GAS", length=100)  
 public String getNhovgOperatingHourAtNoLoadBurningGas() {  
 return this.nhovgOperatingHourAtNoLoadBurningGas;  
 }  
   
 public void setNhovgOperatingHourAtNoLoadBurningGas(String nhovgOperatingHourAtNoLoadBurningGas) {  
 this.nhovgOperatingHourAtNoLoadBurningGas = nhovgOperatingHourAtNoLoadBurningGas;  
 }  
  
   
 @Column(name="NHOVD\_OPERATING\_HOUR\_AT\_NO\_LOAD\_BURNING\_OIL", length=100)  
 public String getNhovdOperatingHourAtNoLoadBurningOil() {  
 return this.nhovdOperatingHourAtNoLoadBurningOil;  
 }  
   
 public void setNhovdOperatingHourAtNoLoadBurningOil(String nhovdOperatingHourAtNoLoadBurningOil) {  
 this.nhovdOperatingHourAtNoLoadBurningOil = nhovdOperatingHourAtNoLoadBurningOil;  
 }  
  
   
 @Column(name="GT1\_OPERATING\_HOUR", length=100)  
 public String getGt1OperatingHour() {  
 return this.gt1OperatingHour;  
 }  
   
 public void setGt1OperatingHour(String gt1OperatingHour) {  
 this.gt1OperatingHour = gt1OperatingHour;  
 }  
  
   
 @Column(name="GT2\_OPERATING\_HOUR", length=100)  
 public String getGt2OperatingHour() {  
 return this.gt2OperatingHour;  
 }  
   
 public void setGt2OperatingHour(String gt2OperatingHour) {  
 this.gt2OperatingHour = gt2OperatingHour;  
 }  
  
   
 @Column(name="NET\_OUTPUT\_CORRECTED\_TO\_SUMMER\_DESIGN\_CFE", length=100)  
 public String getNetOutputCorrectedToSummerDesignCfe() {  
 return this.netOutputCorrectedToSummerDesignCfe;  
 }  
   
 public void setNetOutputCorrectedToSummerDesignCfe(String netOutputCorrectedToSummerDesignCfe) {  
 this.netOutputCorrectedToSummerDesignCfe = netOutputCorrectedToSummerDesignCfe;  
 }  
  
   
 @Column(name="CN\_GAS", length=100)  
 public String getCnGas() {  
 return this.cnGas;  
 }  
   
 public void setCnGas(String cnGas) {  
 this.cnGas = cnGas;  
 }  
  
   
 @Column(name="CN\_OIL", length=100)  
 public String getCnOil() {  
 return this.cnOil;  
 }  
   
 public void setCnOil(String cnOil) {  
 this.cnOil = cnOil;  
 }  
  
   
 @Column(name="SPARE2", length=100)  
 public String getSpare2() {  
 return this.spare2;  
 }  
   
 public void setSpare2(String spare2) {  
 this.spare2 = spare2;  
 }  
  
   
 @Column(name="GT1\_52G\_ON", length=100)  
 public String getGt152gOn() {  
 return this.gt152gOn;  
 }  
   
 public void setGt152gOn(String gt152gOn) {  
 this.gt152gOn = gt152gOn;  
 }  
  
   
 @Column(name="GT2\_52G\_ON", length=100)  
 public String getGt252gOn() {  
 return this.gt252gOn;  
 }  
   
 public void setGt252gOn(String gt252gOn) {  
 this.gt252gOn = gt252gOn;  
 }  
  
   
 @Column(name="E\_A1010\_400KV\_S\_S\_CB\_GT1\_ABN", length=100)  
 public String getEA1010400kvSSCbGt1Abn() {  
 return this.EA1010400kvSSCbGt1Abn;  
 }  
   
 public void setEA1010400kvSSCbGt1Abn(String EA1010400kvSSCbGt1Abn) {  
 this.EA1010400kvSSCbGt1Abn = EA1010400kvSSCbGt1Abn;  
 }  
  
   
 @Column(name="E\_A1010\_400KV\_S\_S\_CB\_GT1\_CLOSE", length=100)  
 public String getEA1010400kvSSCbGt1Close() {  
 return this.EA1010400kvSSCbGt1Close;  
 }  
   
 public void setEA1010400kvSSCbGt1Close(String EA1010400kvSSCbGt1Close) {  
 this.EA1010400kvSSCbGt1Close = EA1010400kvSSCbGt1Close;  
 }  
  
   
 @Column(name="E\_A1010\_400KV\_S\_S\_CB\_GT1\_OPEN", length=100)  
 public String getEA1010400kvSSCbGt1Open() {  
 return this.EA1010400kvSSCbGt1Open;  
 }  
   
 public void setEA1010400kvSSCbGt1Open(String EA1010400kvSSCbGt1Open) {  
 this.EA1010400kvSSCbGt1Open = EA1010400kvSSCbGt1Open;  
 }  
  
   
 @Column(name="E\_A1020\_400KV\_S\_S\_CB\_GT2\_ABN", length=100)  
 public String getEA1020400kvSSCbGt2Abn() {  
 return this.EA1020400kvSSCbGt2Abn;  
 }  
   
 public void setEA1020400kvSSCbGt2Abn(String EA1020400kvSSCbGt2Abn) {  
 this.EA1020400kvSSCbGt2Abn = EA1020400kvSSCbGt2Abn;  
 }  
  
   
 @Column(name="E\_A1020\_400KV\_S\_S\_CB\_GT2\_CLOSE", length=100)  
 public String getEA1020400kvSSCbGt2Close() {  
 return this.EA1020400kvSSCbGt2Close;  
 }  
   
 public void setEA1020400kvSSCbGt2Close(String EA1020400kvSSCbGt2Close) {  
 this.EA1020400kvSSCbGt2Close = EA1020400kvSSCbGt2Close;  
 }  
  
   
 @Column(name="E\_A1020\_400KV\_S\_S\_CB\_GT2\_OPEN", length=100)  
 public String getEA1020400kvSSCbGt2Open() {  
 return this.EA1020400kvSSCbGt2Open;  
 }  
   
 public void setEA1020400kvSSCbGt2Open(String EA1020400kvSSCbGt2Open) {  
 this.EA1020400kvSSCbGt2Open = EA1020400kvSSCbGt2Open;  
 }  
  
   
 @Column(name="E\_A1030\_400KV\_S\_S\_CB\_ST\_CLOSE", length=100)  
 public String getEA1030400kvSSCbStClose() {  
 return this.EA1030400kvSSCbStClose;  
 }  
   
 public void setEA1030400kvSSCbStClose(String EA1030400kvSSCbStClose) {  
 this.EA1030400kvSSCbStClose = EA1030400kvSSCbStClose;  
 }  
  
   
 @Column(name="E\_A1030\_400KV\_S\_S\_CB\_ST\_OPEN", length=100)  
 public String getEA1030400kvSSCbStOpen() {  
 return this.EA1030400kvSSCbStOpen;  
 }  
   
 public void setEA1030400kvSSCbStOpen(String EA1030400kvSSCbStOpen) {  
 this.EA1030400kvSSCbStOpen = EA1030400kvSSCbStOpen;  
 }  
  
   
 @Column(name="GT1\_SYNCHRONIZE", length=100)  
 public String getGt1Synchronize() {  
 return this.gt1Synchronize;  
 }  
   
 public void setGt1Synchronize(String gt1Synchronize) {  
 this.gt1Synchronize = gt1Synchronize;  
 }  
  
   
 @Column(name="GT2\_SYNCHRONIZE", length=100)  
 public String getGt2Synchronize() {  
 return this.gt2Synchronize;  
 }  
   
 public void setGt2Synchronize(String gt2Synchronize) {  
 this.gt2Synchronize = gt2Synchronize;  
 }  
  
   
 @Column(name="ALL\_TARIFF\_METER\_ABNORMAL", length=100)  
 public String getAllTariffMeterAbnormal() {  
 return this.allTariffMeterAbnormal;  
 }  
   
 public void setAllTariffMeterAbnormal(String allTariffMeterAbnormal) {  
 this.allTariffMeterAbnormal = allTariffMeterAbnormal;  
 }  
  
   
 @Column(name="GT1\_TRANSMITTER\_WAT\_AND\_VAR\_ABNORMAL", length=100)  
 public String getGt1TransmitterWatAndVarAbnormal() {  
 return this.gt1TransmitterWatAndVarAbnormal;  
 }  
   
 public void setGt1TransmitterWatAndVarAbnormal(String gt1TransmitterWatAndVarAbnormal) {  
 this.gt1TransmitterWatAndVarAbnormal = gt1TransmitterWatAndVarAbnormal;  
 }  
  
   
 @Column(name="GT2\_TRANSMITTER\_WAT\_AND\_VAR\_ABNORMAL", length=100)  
 public String getGt2TransmitterWatAndVarAbnormal() {  
 return this.gt2TransmitterWatAndVarAbnormal;  
 }  
   
 public void setGt2TransmitterWatAndVarAbnormal(String gt2TransmitterWatAndVarAbnormal) {  
 this.gt2TransmitterWatAndVarAbnormal = gt2TransmitterWatAndVarAbnormal;  
 }  
  
   
 @Column(name="ST\_TRANSMITTER\_WAT\_AND\_VAR\_ABNORMAL", length=100)  
 public String getStTransmitterWatAndVarAbnormal() {  
 return this.stTransmitterWatAndVarAbnormal;  
 }  
   
 public void setStTransmitterWatAndVarAbnormal(String stTransmitterWatAndVarAbnormal) {  
 this.stTransmitterWatAndVarAbnormal = stTransmitterWatAndVarAbnormal;  
 }  
  
   
 @Column(name="NO1\_TEMP\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo1TempTransmitterAbnormal() {  
 return this.no1TempTransmitterAbnormal;  
 }  
   
 public void setNo1TempTransmitterAbnormal(String no1TempTransmitterAbnormal) {  
 this.no1TempTransmitterAbnormal = no1TempTransmitterAbnormal;  
 }  
  
   
 @Column(name="NO2\_TEMP\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo2TempTransmitterAbnormal() {  
 return this.no2TempTransmitterAbnormal;  
 }  
   
 public void setNo2TempTransmitterAbnormal(String no2TempTransmitterAbnormal) {  
 this.no2TempTransmitterAbnormal = no2TempTransmitterAbnormal;  
 }  
  
   
 @Column(name="TEMP\_TRANSMITTER\_SELECT\_0\_NO2\_1\_NO1", length=100)  
 public String getTempTransmitterSelect0No21No1() {  
 return this.tempTransmitterSelect0No21No1;  
 }  
   
 public void setTempTransmitterSelect0No21No1(String tempTransmitterSelect0No21No1) {  
 this.tempTransmitterSelect0No21No1 = tempTransmitterSelect0No21No1;  
 }  
  
   
 @Column(name="NO1\_HUMIDITY\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo1HumidityTransmitterAbnormal() {  
 return this.no1HumidityTransmitterAbnormal;  
 }  
   
 public void setNo1HumidityTransmitterAbnormal(String no1HumidityTransmitterAbnormal) {  
 this.no1HumidityTransmitterAbnormal = no1HumidityTransmitterAbnormal;  
 }  
  
   
 @Column(name="NO2\_HUMIDITY\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo2HumidityTransmitterAbnormal() {  
 return this.no2HumidityTransmitterAbnormal;  
 }  
   
 public void setNo2HumidityTransmitterAbnormal(String no2HumidityTransmitterAbnormal) {  
 this.no2HumidityTransmitterAbnormal = no2HumidityTransmitterAbnormal;  
 }  
  
   
 @Column(name="HUMIDITY\_TRANSMITTER\_SELECT\_0\_NO2\_1\_NO1", length=100)  
 public String getHumidityTransmitterSelect0No21No1() {  
 return this.humidityTransmitterSelect0No21No1;  
 }  
   
 public void setHumidityTransmitterSelect0No21No1(String humidityTransmitterSelect0No21No1) {  
 this.humidityTransmitterSelect0No21No1 = humidityTransmitterSelect0No21No1;  
 }  
  
   
 @Column(name="NO1\_AMBIENT\_PRESSURE\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo1AmbientPressureTransmitterAbnormal() {  
 return this.no1AmbientPressureTransmitterAbnormal;  
 }  
   
 public void setNo1AmbientPressureTransmitterAbnormal(String no1AmbientPressureTransmitterAbnormal) {  
 this.no1AmbientPressureTransmitterAbnormal = no1AmbientPressureTransmitterAbnormal;  
 }  
  
   
 @Column(name="NO2\_AMBIENT\_PRESSURE\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo2AmbientPressureTransmitterAbnormal() {  
 return this.no2AmbientPressureTransmitterAbnormal;  
 }  
   
 public void setNo2AmbientPressureTransmitterAbnormal(String no2AmbientPressureTransmitterAbnormal) {  
 this.no2AmbientPressureTransmitterAbnormal = no2AmbientPressureTransmitterAbnormal;  
 }  
  
   
 @Column(name="AMBIENT\_PRESSURE\_TRANSMITTER\_SELECT\_0\_NO2\_1\_NO1", length=100)  
 public String getAmbientPressureTransmitterSelect0No21No1() {  
 return this.ambientPressureTransmitterSelect0No21No1;  
 }  
   
 public void setAmbientPressureTransmitterSelect0No21No1(String ambientPressureTransmitterSelect0No21No1) {  
 this.ambientPressureTransmitterSelect0No21No1 = ambientPressureTransmitterSelect0No21No1;  
 }  
  
   
 @Column(name="NO1\_SEA\_WATER\_TEMPERATURE\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo1SeaWaterTemperatureTransmitterAbnormal() {  
 return this.no1SeaWaterTemperatureTransmitterAbnormal;  
 }  
   
 public void setNo1SeaWaterTemperatureTransmitterAbnormal(String no1SeaWaterTemperatureTransmitterAbnormal) {  
 this.no1SeaWaterTemperatureTransmitterAbnormal = no1SeaWaterTemperatureTransmitterAbnormal;  
 }  
  
   
 @Column(name="NO2\_SEA\_WATER\_TEMPERATURE\_TRANSMITTER\_ABNORMAL", length=100)  
 public String getNo2SeaWaterTemperatureTransmitterAbnormal() {  
 return this.no2SeaWaterTemperatureTransmitterAbnormal;  
 }  
   
 public void setNo2SeaWaterTemperatureTransmitterAbnormal(String no2SeaWaterTemperatureTransmitterAbnormal) {  
 this.no2SeaWaterTemperatureTransmitterAbnormal = no2SeaWaterTemperatureTransmitterAbnormal;  
 }  
  
   
 @Column(name="SEA\_WATER\_TEMPERATURE\_TRANSMITTER\_SELECT\_0\_NO2\_1\_NO1", length=100)  
 public String getSeaWaterTemperatureTransmitterSelect0No21No1() {  
 return this.seaWaterTemperatureTransmitterSelect0No21No1;  
 }  
   
 public void setSeaWaterTemperatureTransmitterSelect0No21No1(String seaWaterTemperatureTransmitterSelect0No21No1) {  
 this.seaWaterTemperatureTransmitterSelect0No21No1 = seaWaterTemperatureTransmitterSelect0No21No1;  
 }  
  
   
 @Column(name="GT1\_GAS\_SELECT", length=100)  
 public String getGt1GasSelect() {  
 return this.gt1GasSelect;  
 }  
   
 public void setGt1GasSelect(String gt1GasSelect) {  
 this.gt1GasSelect = gt1GasSelect;  
 }  
  
   
 @Column(name="GT2\_GAS\_SELECT", length=100)  
 public String getGt2GasSelect() {  
 return this.gt2GasSelect;  
 }  
   
 public void setGt2GasSelect(String gt2GasSelect) {  
 this.gt2GasSelect = gt2GasSelect;  
 }  
  
   
 @Column(name="GT1\_OIL\_SELECT", length=100)  
 public String getGt1OilSelect() {  
 return this.gt1OilSelect;  
 }  
   
 public void setGt1OilSelect(String gt1OilSelect) {  
 this.gt1OilSelect = gt1OilSelect;  
 }  
  
   
 @Column(name="GT2\_OIL\_SELECT", length=100)  
 public String getGt2OilSelect() {  
 return this.gt2OilSelect;  
 }  
   
 public void setGt2OilSelect(String gt2OilSelect) {  
 this.gt2OilSelect = gt2OilSelect;  
 }  
  
   
 @Column(name="NO1\_GT\_FCB\_SELECT", length=100)  
 public String getNo1GtFcbSelect() {  
 return this.no1GtFcbSelect;  
 }  
   
 public void setNo1GtFcbSelect(String no1GtFcbSelect) {  
 this.no1GtFcbSelect = no1GtFcbSelect;  
 }  
  
   
 @Column(name="NO2\_GT\_FCB\_SELECT", length=100)  
 public String getNo2GtFcbSelect() {  
 return this.no2GtFcbSelect;  
 }  
   
 public void setNo2GtFcbSelect(String no2GtFcbSelect) {  
 this.no2GtFcbSelect = no2GtFcbSelect;  
 }  
  
   
 @Column(name="FCB\_COMMAND", length=100)  
 public String getFcbCommand() {  
 return this.fcbCommand;  
 }  
   
 public void setFcbCommand(String fcbCommand) {  
 this.fcbCommand = fcbCommand;  
 }  
  
   
 @Column(name="GT1\_GAS\_COMB", length=100)  
 public String getGt1GasComb() {  
 return this.gt1GasComb;  
 }  
   
 public void setGt1GasComb(String gt1GasComb) {  
 this.gt1GasComb = gt1GasComb;  
 }  
  
   
 @Column(name="GT2\_GAS\_COMB", length=100)  
 public String getGt2GasComb() {  
 return this.gt2GasComb;  
 }  
   
 public void setGt2GasComb(String gt2GasComb) {  
 this.gt2GasComb = gt2GasComb;  
 }  
  
   
 @Column(name="GT1\_OIL\_COMB", length=100)  
 public String getGt1OilComb() {  
 return this.gt1OilComb;  
 }  
   
 public void setGt1OilComb(String gt1OilComb) {  
 this.gt1OilComb = gt1OilComb;  
 }  
  
   
 @Column(name="GT2\_OIL\_COMB", length=100)  
 public String getGt2OilComb() {  
 return this.gt2OilComb;  
 }  
   
 public void setGt2OilComb(String gt2OilComb) {  
 this.gt2OilComb = gt2OilComb;  
 }  
  
   
 @Column(name="GT1\_ENCENDER\_COMB", length=100)  
 public String getGt1EncenderComb() {  
 return this.gt1EncenderComb;  
 }  
   
 public void setGt1EncenderComb(String gt1EncenderComb) {  
 this.gt1EncenderComb = gt1EncenderComb;  
 }  
  
   
 @Column(name="GT2\_ENCENDER\_COMB", length=100)  
 public String getGt2EncenderComb() {  
 return this.gt2EncenderComb;  
 }  
   
 public void setGt2EncenderComb(String gt2EncenderComb) {  
 this.gt2EncenderComb = gt2EncenderComb;  
 }  
  
   
 @Column(name="ACTUAL\_HIGHER\_HEATING\_VALUE\_WET", length=100)  
 public String getActualHigherHeatingValueWet() {  
 return this.actualHigherHeatingValueWet;  
 }  
   
 public void setActualHigherHeatingValueWet(String actualHigherHeatingValueWet) {  
 this.actualHigherHeatingValueWet = actualHigherHeatingValueWet;  
 }  
  
   
 @Column(name="ACTUAL\_LOWER\_HEATING\_VALUE\_WET", length=100)  
 public String getActualLowerHeatingValueWet() {  
 return this.actualLowerHeatingValueWet;  
 }  
   
 public void setActualLowerHeatingValueWet(String actualLowerHeatingValueWet) {  
 this.actualLowerHeatingValueWet = actualLowerHeatingValueWet;  
 }  
  
   
 @Column(name="ACTUAL\_HIGHER\_HEATING\_VALUE\_WET2", length=100)  
 public String getActualHigherHeatingValueWet2() {  
 return this.actualHigherHeatingValueWet2;  
 }  
   
 public void setActualHigherHeatingValueWet2(String actualHigherHeatingValueWet2) {  
 this.actualHigherHeatingValueWet2 = actualHigherHeatingValueWet2;  
 }  
  
   
 @Column(name="ACTUAL\_LOWER\_HEATING\_VALUE\_WET2", length=100)  
 public String getActualLowerHeatingValueWet2() {  
 return this.actualLowerHeatingValueWet2;  
 }  
   
 public void setActualLowerHeatingValueWet2(String actualLowerHeatingValueWet2) {  
 this.actualLowerHeatingValueWet2 = actualLowerHeatingValueWet2;  
 }  
  
   
 @Column(name="GAS\_VOLUME\_TOTAL", length=100)  
 public String getGasVolumeTotal() {  
 return this.gasVolumeTotal;  
 }  
   
 public void setGasVolumeTotal(String gasVolumeTotal) {  
 this.gasVolumeTotal = gasVolumeTotal;  
 }  
  
   
 @Column(name="GAS\_MASS\_TOTAL", length=100)  
 public String getGasMassTotal() {  
 return this.gasMassTotal;  
 }  
   
 public void setGasMassTotal(String gasMassTotal) {  
 this.gasMassTotal = gasMassTotal;  
 }  
  
   
 @Column(name="QG5\_5MIN\_HEAT\_INPUT\_WITH\_GAS", length=100)  
 public String getQg55minHeatInputWithGas() {  
 return this.qg55minHeatInputWithGas;  
 }  
   
 public void setQg55minHeatInputWithGas(String qg55minHeatInputWithGas) {  
 this.qg55minHeatInputWithGas = qg55minHeatInputWithGas;  
 }  
  
   
 @Column(name="QD5\_5MIN\_HEAT\_INPUT\_WITH\_OIL", length=100)  
 public String getQd55minHeatInputWithOil() {  
 return this.qd55minHeatInputWithOil;  
 }  
   
 public void setQd55minHeatInputWithOil(String qd55minHeatInputWithOil) {  
 this.qd55minHeatInputWithOil = qd55minHeatInputWithOil;  
 }  
  
   
 @Column(name="GT1\_OPERATION\_MODE", length=100)  
 public String getGt1OperationMode() {  
 return this.gt1OperationMode;  
 }  
   
 public void setGt1OperationMode(String gt1OperationMode) {  
 this.gt1OperationMode = gt1OperationMode;  
 }  
  
   
 @Column(name="GT2\_OPERATION\_MODE", length=100)  
 public String getGt2OperationMode() {  
 return this.gt2OperationMode;  
 }  
   
 public void setGt2OperationMode(String gt2OperationMode) {  
 this.gt2OperationMode = gt2OperationMode;  
 }  
  
   
 @Column(name="LINE1\_WATT\_HOUR\_MAS\_5MIN\_IN\_USE", length=100)  
 public String getLine1WattHourMas5minInUse() {  
 return this.line1WattHourMas5minInUse;  
 }  
   
 public void setLine1WattHourMas5minInUse(String line1WattHourMas5minInUse) {  
 this.line1WattHourMas5minInUse = line1WattHourMas5minInUse;  
 }  
  
   
 @Column(name="LINE2\_WATT\_HOUR\_MAS\_5MIN\_IN\_USE", length=100)  
 public String getLine2WattHourMas5minInUse() {  
 return this.line2WattHourMas5minInUse;  
 }  
   
 public void setLine2WattHourMas5minInUse(String line2WattHourMas5minInUse) {  
 this.line2WattHourMas5minInUse = line2WattHourMas5minInUse;  
 }  
  
   
 @Column(name="CTOV", length=100)  
 public String getCtov() {  
 return this.ctov;  
 }  
   
 public void setCtov(String ctov) {  
 this.ctov = ctov;  
 }  
  
   
 @Column(name="LINE1\_VAR\_HOUR\_5MIN\_KVARH", length=100)  
 public String getLine1VarHour5minKvarh() {  
 return this.line1VarHour5minKvarh;  
 }  
   
 public void setLine1VarHour5minKvarh(String line1VarHour5minKvarh) {  
 this.line1VarHour5minKvarh = line1VarHour5minKvarh;  
 }  
  
   
 @Column(name="LINE2\_VAR\_HOUR\_5MIN\_KVARH", length=100)  
 public String getLine2VarHour5minKvarh() {  
 return this.line2VarHour5minKvarh;  
 }  
   
 public void setLine2VarHour5minKvarh(String line2VarHour5minKvarh) {  
 this.line2VarHour5minKvarh = line2VarHour5minKvarh;  
 }  
  
   
 @Column(name="LINE1\_VAR\_HOUR\_5MIN\_KVARH\_BACKUP", length=100)  
 public String getLine1VarHour5minKvarhBackup() {  
 return this.line1VarHour5minKvarhBackup;  
 }  
   
 public void setLine1VarHour5minKvarhBackup(String line1VarHour5minKvarhBackup) {  
 this.line1VarHour5minKvarhBackup = line1VarHour5minKvarhBackup;  
 }  
  
   
 @Column(name="LINE2\_VAR\_HOUR\_5MIN\_KVARH\_BACKUP", length=100)  
 public String getLine2VarHour5minKvarhBackup() {  
 return this.line2VarHour5minKvarhBackup;  
 }  
   
 public void setLine2VarHour5minKvarhBackup(String line2VarHour5minKvarhBackup) {  
 this.line2VarHour5minKvarhBackup = line2VarHour5minKvarhBackup;  
 }  
  
   
 @Column(name="CN", length=100)  
 public String getCn() {  
 return this.cn;  
 }  
   
 public void setCn(String cn) {  
 this.cn = cn;  
 }  
  
   
 @Column(name="CNG100CORR\_CNG", length=100)  
 public String getCng100corrCng() {  
 return this.cng100corrCng;  
 }  
   
 public void setCng100corrCng(String cng100corrCng) {  
 this.cng100corrCng = cng100corrCng;  
 }  
  
   
 @Column(name="LINE1\_WATT\_HOUR\_MENOS\_5MIN\_IN\_USE", length=100)  
 public String getLine1WattHourMenos5minInUse() {  
 return this.line1WattHourMenos5minInUse;  
 }  
   
 public void setLine1WattHourMenos5minInUse(String line1WattHourMenos5minInUse) {  
 this.line1WattHourMenos5minInUse = line1WattHourMenos5minInUse;  
 }  
  
   
 @Column(name="LINE2\_WATT\_HOUR\_MENOS\_5MIN\_IN\_USE", length=100)  
 public String getLine2WattHourMenos5minInUse() {  
 return this.line2WattHourMenos5minInUse;  
 }  
   
 public void setLine2WattHourMenos5minInUse(String line2WattHourMenos5minInUse) {  
 this.line2WattHourMenos5minInUse = line2WattHourMenos5minInUse;  
 }  
  
   
 @Column(name="KILOVARHOUR\_PREVIOUS", length=100)  
 public String getKilovarhourPrevious() {  
 return this.kilovarhourPrevious;  
 }  
   
 public void setKilovarhourPrevious(String kilovarhourPrevious) {  
 this.kilovarhourPrevious = kilovarhourPrevious;  
 }  
  
   
 @Column(name="LINE1\_KILOVARHOUR\_PREVIOUS\_PROGQ1\_IN\_USE", length=100)  
 public String getLine1KilovarhourPreviousProgq1InUse() {  
 return this.line1KilovarhourPreviousProgq1InUse;  
 }  
   
 public void setLine1KilovarhourPreviousProgq1InUse(String line1KilovarhourPreviousProgq1InUse) {  
 this.line1KilovarhourPreviousProgq1InUse = line1KilovarhourPreviousProgq1InUse;  
 }  
  
   
 @Column(name="LINE1\_KILOVARHOUR\_PREVIOUS\_PROGQ4\_IN\_USE", length=100)  
 public String getLine1KilovarhourPreviousProgq4InUse() {  
 return this.line1KilovarhourPreviousProgq4InUse;  
 }  
   
 public void setLine1KilovarhourPreviousProgq4InUse(String line1KilovarhourPreviousProgq4InUse) {  
 this.line1KilovarhourPreviousProgq4InUse = line1KilovarhourPreviousProgq4InUse;  
 }  
  
   
 @Column(name="LINE2\_KILOVARHOUR\_PREVIOUS\_PROGQ1\_IN\_USE", length=100)  
 public String getLine2KilovarhourPreviousProgq1InUse() {  
 return this.line2KilovarhourPreviousProgq1InUse;  
 }  
   
 public void setLine2KilovarhourPreviousProgq1InUse(String line2KilovarhourPreviousProgq1InUse) {  
 this.line2KilovarhourPreviousProgq1InUse = line2KilovarhourPreviousProgq1InUse;  
 }  
  
   
 @Column(name="LINE2\_KILOVARHOUR\_PREVIOUS\_PROGQ4\_IN\_USE", length=100)  
 public String getLine2KilovarhourPreviousProgq4InUse() {  
 return this.line2KilovarhourPreviousProgq4InUse;  
 }  
   
 public void setLine2KilovarhourPreviousProgq4InUse(String line2KilovarhourPreviousProgq4InUse) {  
 this.line2KilovarhourPreviousProgq4InUse = line2KilovarhourPreviousProgq4InUse;  
 }  
  
   
 @Column(name="LINE1\_KILOVARHOUR\_PREVIOUS\_PROGQ1\_MAIN", length=100)  
 public String getLine1KilovarhourPreviousProgq1Main() {  
 return this.line1KilovarhourPreviousProgq1Main;  
 }  
   
 public void setLine1KilovarhourPreviousProgq1Main(String line1KilovarhourPreviousProgq1Main) {  
 this.line1KilovarhourPreviousProgq1Main = line1KilovarhourPreviousProgq1Main;  
 }  
  
   
 @Column(name="LINE1\_KILOVARHOUR\_PREVIOUS\_PROGQ4\_MAIN", length=100)  
 public String getLine1KilovarhourPreviousProgq4Main() {  
 return this.line1KilovarhourPreviousProgq4Main;  
 }  
   
 public void setLine1KilovarhourPreviousProgq4Main(String line1KilovarhourPreviousProgq4Main) {  
 this.line1KilovarhourPreviousProgq4Main = line1KilovarhourPreviousProgq4Main;  
 }  
  
   
 @Column(name="LINE2\_KILOVARHOUR\_PREVIOUS\_PROGQ1\_MAIN", length=100)  
 public String getLine2KilovarhourPreviousProgq1Main() {  
 return this.line2KilovarhourPreviousProgq1Main;  
 }  
   
 public void setLine2KilovarhourPreviousProgq1Main(String line2KilovarhourPreviousProgq1Main) {  
 this.line2KilovarhourPreviousProgq1Main = line2KilovarhourPreviousProgq1Main;  
 }  
  
   
 @Column(name="LINE2\_KILOVARHOUR\_PREVIOUS\_PROGQ4\_MAIN", length=100)  
 public String getLine2KilovarhourPreviousProgq4Main() {  
 return this.line2KilovarhourPreviousProgq4Main;  
 }  
   
 public void setLine2KilovarhourPreviousProgq4Main(String line2KilovarhourPreviousProgq4Main) {  
 this.line2KilovarhourPreviousProgq4Main = line2KilovarhourPreviousProgq4Main;  
 }  
  
   
 @Column(name="LINE1\_KILOVARHOUR\_PREVIOUS\_PROGQ1\_BACKUP", length=100)  
 public String getLine1KilovarhourPreviousProgq1Backup() {  
 return this.line1KilovarhourPreviousProgq1Backup;  
 }  
   
 public void setLine1KilovarhourPreviousProgq1Backup(String line1KilovarhourPreviousProgq1Backup) {  
 this.line1KilovarhourPreviousProgq1Backup = line1KilovarhourPreviousProgq1Backup;  
 }  
  
   
 @Column(name="LINE1\_KILOVARHOUR\_PREVIOUS\_PROGQ4\_BACKUP", length=100)  
 public String getLine1KilovarhourPreviousProgq4Backup() {  
 return this.line1KilovarhourPreviousProgq4Backup;  
 }  
   
 public void setLine1KilovarhourPreviousProgq4Backup(String line1KilovarhourPreviousProgq4Backup) {  
 this.line1KilovarhourPreviousProgq4Backup = line1KilovarhourPreviousProgq4Backup;  
 }  
  
   
 @Column(name="LINE2\_KILOVARHOUR\_PREVIOUS\_PROGQ1\_BACKUP", length=100)  
 public String getLine2KilovarhourPreviousProgq1Backup() {  
 return this.line2KilovarhourPreviousProgq1Backup;  
 }  
   
 public void setLine2KilovarhourPreviousProgq1Backup(String line2KilovarhourPreviousProgq1Backup) {  
 this.line2KilovarhourPreviousProgq1Backup = line2KilovarhourPreviousProgq1Backup;  
 }  
  
   
 @Column(name="LINE2\_KILOVARHOUR\_PREVIOUS\_PROGQ4\_BACKUP", length=100)  
 public String getLine2KilovarhourPreviousProgq4Backup() {  
 return this.line2KilovarhourPreviousProgq4Backup;  
 }  
   
 public void setLine2KilovarhourPreviousProgq4Backup(String line2KilovarhourPreviousProgq4Backup) {  
 this.line2KilovarhourPreviousProgq4Backup = line2KilovarhourPreviousProgq4Backup;  
 }  
  
   
 @Column(name="TIPO\_CARGA", length=100)  
 public String getTipoCarga() {  
 return this.tipoCarga;  
 }  
   
 public void setTipoCarga(String tipoCarga) {  
 this.tipoCarga = tipoCarga;  
 }  
  
   
 @Column(name="CARGA\_ID", precision=22, scale=0)  
 public BigDecimal getCargaId() {  
 return this.cargaId;  
 }  
   
 public void setCargaId(BigDecimal cargaId) {  
 this.cargaId = cargaId;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="HORA\_CARGA", nullable=false, length=11)  
 public Date getHoraCarga() {  
 return this.horaCarga;  
 }  
   
 public void setHoraCarga(Date horaCarga) {  
 this.horaCarga = horaCarga;  
 }  
  
   
 @Column(name="ESTATUS\_LIMPIO", nullable=false, precision=38, scale=0)  
 public BigDecimal getEstatusLimpio() {  
 return this.estatusLimpio;  
 }  
   
 public void setEstatusLimpio(BigDecimal estatusLimpio) {  
 this.estatusLimpio = estatusLimpio;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxDocumentos.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import java.util.Date;  
import java.util.HashSet;  
import java.util.Set;  
  
import javax.persistence.CascadeType;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* TxDocumentos generated by hbm2java  
 \*/  
@Entity  
@Table(name = "TX\_DOCUMENTOS")  
public class TxDocumentos implements java.io.Serializable {  
  
  
 private BigDecimal documentoId;  
 private CatTipoDocumento catTipoDocumento;  
 private TxEscenario txEscenario;  
 private TxUsuario txUsuario;  
 private CatCompaniaPlantas catCompaniaPlantas;  
 private CatEstatusFactura catEstatusFactura;  
 private CatTiposFacturas catTiposFacturas;  
 private CatTipoRelacion catTipoRelacion;  
 private CatClientes catClientes;  
 private CatUsoCfdi catUsoCfdi;  
 private CatFormasPago catFormasPago;  
 private CatMonedas catMonedas;  
 private CatEstatusPagos catEstatusPagos;  
 private CatMetodoPago catMetodoPago;  
 private CatVersion catVersion;  
 private BigDecimal usuarioIdCancelo;  
 private Date fechaHoraCreacion;  
 private Date fechaCierre;  
 private String serie;  
 private String folio;  
 private Date fechaHoraSellado;  
 private String uuidSat;  
 private BigDecimal diasCredito;  
 private Date fechaRevision;  
 private Date fechaVencimiento;  
 private String cuenta;  
 private String confirmacionSat;  
 private String condicionesPago;  
 private String selloDigitalEmisor;  
 private String selloDigitaSat;  
 private String cadenaOriginal;  
 private String timbrado;  
 private String certificado;  
 private String noCertificado;  
 private String noCertificadoSat;  
 private String strXml;  
 private String strXmlCancelacion;  
 private Date fechaHoraCancelacion;  
 private String observaciones;  
 private Set<TxFacturaUuids> txFacturaUuidsesForDocumentoId = new HashSet<TxFacturaUuids>(0);  
 private Set<CatFacturaProductos> catFacturaProductoses = new HashSet<CatFacturaProductos>(0);  
 private Set<TxFacturaUuids> txFacturaUuidsesForDocumentoAsociadoId = new HashSet<TxFacturaUuids>(0);  
  
 public TxDocumentos() {  
 }  
  
 public TxDocumentos(CatTipoDocumento catTipoDocumento, TxEscenario txEscenario, TxUsuario txUsuario,  
 CatCompaniaPlantas catCompaniaPlantas, CatEstatusFactura catEstatusFactura,  
 CatTiposFacturas catTiposFacturas, CatTipoRelacion catTipoRelacion, CatClientes catClientes,  
 CatUsoCfdi catUsoCfdi, CatFormasPago catFormasPago, CatMonedas catMonedas,  
 CatEstatusPagos catEstatusPagos, CatMetodoPago catMetodoPago, CatVersion catVersion,  
 BigDecimal usuarioIdCancelo, Date fechaHoraCreacion, Date fechaCierre, String serie,  
 String folio, Date fechaHoraSellado, String uuidSat, BigDecimal diasCredito, Date fechaRevision,  
 Date fechaVencimiento, String cuenta, String confirmacionSat, String condicionesPago,  
 String selloDigitalEmisor, String selloDigitaSat, String cadenaOriginal, String timbrado,  
 String certificado, String noCertificado, String strXml, String strXmlCancelacion,  
 Date fechaHoraCancelacion, String observaciones,  
 Set<TxFacturaUuids> txFacturaUuidsesForDocumentoId,  
 Set<CatFacturaProductos> catFacturaProductoses,  
 Set<TxFacturaUuids> txFacturaUuidsesForDocumentoAsociadoId) {  
 this.catTipoDocumento = catTipoDocumento;  
 this.txEscenario = txEscenario;  
 this.txUsuario = txUsuario;  
 this.catCompaniaPlantas = catCompaniaPlantas;  
 this.catEstatusFactura = catEstatusFactura;  
 this.catTiposFacturas = catTiposFacturas;  
 this.catTipoRelacion = catTipoRelacion;  
 this.catClientes = catClientes;  
 this.catUsoCfdi = catUsoCfdi;  
 this.catFormasPago = catFormasPago;  
 this.catMonedas = catMonedas;  
 this.catEstatusPagos = catEstatusPagos;  
 this.catMetodoPago = catMetodoPago;  
 this.catVersion = catVersion;  
 this.usuarioIdCancelo = usuarioIdCancelo;  
 this.fechaHoraCreacion = fechaHoraCreacion;  
 this.fechaCierre = fechaCierre;  
 this.serie = serie;  
 this.folio = folio;  
 this.fechaHoraSellado = fechaHoraSellado;  
 this.uuidSat = uuidSat;  
 this.diasCredito = diasCredito;  
 this.fechaRevision = fechaRevision;  
 this.fechaVencimiento = fechaVencimiento;  
 this.cuenta = cuenta;  
 this.confirmacionSat = confirmacionSat;  
 this.condicionesPago = condicionesPago;  
 this.selloDigitalEmisor = selloDigitalEmisor;  
 this.selloDigitaSat = selloDigitaSat;  
 this.cadenaOriginal = cadenaOriginal;  
 this.timbrado = timbrado;  
 this.certificado = certificado;  
 this.noCertificado = noCertificado;  
 this.strXml = strXml;  
 this.strXmlCancelacion = strXmlCancelacion;  
 this.fechaHoraCancelacion = fechaHoraCancelacion;  
 this.observaciones = observaciones;  
 this.txFacturaUuidsesForDocumentoId = txFacturaUuidsesForDocumentoId;  
 this.catFacturaProductoses = catFacturaProductoses;  
 this.txFacturaUuidsesForDocumentoAsociadoId = txFacturaUuidsesForDocumentoAsociadoId;  
 }  
  
 @Id  
 @GeneratedValue(strategy = IDENTITY)  
  
  
 @Column(name = "DOCUMENTO\_ID", unique = true, nullable = false, precision = 22, scale = 0)  
 public BigDecimal getDocumentoId() {  
 return this.documentoId;  
 }  
  
 public void setDocumentoId(BigDecimal documentoId) {  
 this.documentoId = documentoId;  
 }  
  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "TIPO\_DOCUMENTO\_ID")  
 public CatTipoDocumento getCatTipoDocumento() {  
 return this.catTipoDocumento;  
 }  
  
 public void setCatTipoDocumento(CatTipoDocumento catTipoDocumento) {  
 this.catTipoDocumento = catTipoDocumento;  
 }  
  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "ESCENARIO\_ID")  
 public TxEscenario getTxEscenario() {  
 return this.txEscenario;  
 }  
  
 public void setTxEscenario(TxEscenario txEscenario) {  
 this.txEscenario = txEscenario;  
 }  
  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "USUARIO\_ID\_CREACION")  
 public TxUsuario getTxUsuario() {  
 return this.txUsuario;  
 }  
  
 public void setTxUsuario(TxUsuario txUsuario) {  
 this.txUsuario = txUsuario;  
 }  
  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "COMPANIA\_PLANTA\_ID")  
 public CatCompaniaPlantas getCatCompaniaPlantas() {  
 return this.catCompaniaPlantas;  
 }  
  
 public void setCatCompaniaPlantas(CatCompaniaPlantas catCompaniaPlantas) {  
 this.catCompaniaPlantas = catCompaniaPlantas;  
 }  
  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "ESTATUS\_FACTURA\_ID")  
 public CatEstatusFactura getCatEstatusFactura() {  
 return this.catEstatusFactura;  
 }  
  
 public void setCatEstatusFactura(CatEstatusFactura catEstatusFactura) {  
 this.catEstatusFactura = catEstatusFactura;  
 }  
  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "TIPO\_FACTURA\_ID")  
 public CatTiposFacturas getCatTiposFacturas() {  
 return this.catTiposFacturas;  
 }  
  
 public void setCatTiposFacturas(CatTiposFacturas catTiposFacturas) {  
 this.catTiposFacturas = catTiposFacturas;  
 }  
  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "TIPO\_RELACION\_ID")  
 public CatTipoRelacion getCatTipoRelacion() {  
 return this.catTipoRelacion;  
 }  
  
 public void setCatTipoRelacion(CatTipoRelacion catTipoRelacion) {  
 this.catTipoRelacion = catTipoRelacion;  
 }  
  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "CLIENTE\_ID")  
 public CatClientes getCatClientes() {  
 return this.catClientes;  
 }  
  
 public void setCatClientes(CatClientes catClientes) {  
 this.catClientes = catClientes;  
 }  
  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "USO\_CFDI\_ID")  
 public CatUsoCfdi getCatUsoCfdi() {  
 return this.catUsoCfdi;  
 }  
  
 public void setCatUsoCfdi(CatUsoCfdi catUsoCfdi) {  
 this.catUsoCfdi = catUsoCfdi;  
 }  
  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "FORMA\_PAGO\_ID")  
 public CatFormasPago getCatFormasPago() {  
 return this.catFormasPago;  
 }  
  
 public void setCatFormasPago(CatFormasPago catFormasPago) {  
 this.catFormasPago = catFormasPago;  
 }  
  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "MONEDA\_ID")  
 public CatMonedas getCatMonedas() {  
 return this.catMonedas;  
 }  
  
 public void setCatMonedas(CatMonedas catMonedas) {  
 this.catMonedas = catMonedas;  
 }  
  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "ESTATUS\_PAGO\_ID")  
 public CatEstatusPagos getCatEstatusPagos() {  
 return this.catEstatusPagos;  
 }  
  
 public void setCatEstatusPagos(CatEstatusPagos catEstatusPagos) {  
 this.catEstatusPagos = catEstatusPagos;  
 }  
  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "METODO\_PAGO\_ID")  
 public CatMetodoPago getCatMetodoPago() {  
 return this.catMetodoPago;  
 }  
  
 public void setCatMetodoPago(CatMetodoPago catMetodoPago) {  
 this.catMetodoPago = catMetodoPago;  
 }  
  
 @ManyToOne(fetch = FetchType.LAZY)  
 @JoinColumn(name = "VERSION\_ID")  
 public CatVersion getCatVersion() {  
 return this.catVersion;  
 }  
  
 public void setCatVersion(CatVersion catVersion) {  
 this.catVersion = catVersion;  
 }  
  
  
 @Column(name = "USUARIO\_ID\_CANCELO", precision = 22, scale = 0)  
 public BigDecimal getUsuarioIdCancelo() {  
 return this.usuarioIdCancelo;  
 }  
  
 public void setUsuarioIdCancelo(BigDecimal usuarioIdCancelo) {  
 this.usuarioIdCancelo = usuarioIdCancelo;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name = "FECHA\_HORA\_CREACION", length = 11)  
 public Date getFechaHoraCreacion() {  
 return this.fechaHoraCreacion;  
 }  
  
 public void setFechaHoraCreacion(Date fechaHoraCreacion) {  
 this.fechaHoraCreacion = fechaHoraCreacion;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name = "FECHA\_CIERRE", length = 7)  
 public Date getFechaCierre() {  
 return this.fechaCierre;  
 }  
  
 public void setFechaCierre(Date fechaCierre) {  
 this.fechaCierre = fechaCierre;  
 }  
  
  
 @Column(name = "SERIE")  
 public String getSerie() {  
 return this.serie;  
 }  
  
 public void setSerie(String serie) {  
 this.serie = serie;  
 }  
  
  
 @Column(name = "FOLIO")  
 public String getFolio() {  
 return this.folio;  
 }  
  
 public void setFolio(String folio) {  
 this.folio = folio;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name = "FECHA\_HORA\_SELLADO", length = 11)  
 public Date getFechaHoraSellado() {  
 return this.fechaHoraSellado;  
 }  
  
 public void setFechaHoraSellado(Date fechaHoraSellado) {  
 this.fechaHoraSellado = fechaHoraSellado;  
 }  
  
  
 @Column(name = "UUID\_SAT")  
 public String getUuidSat() {  
 return this.uuidSat;  
 }  
  
 public void setUuidSat(String uuidSat) {  
 this.uuidSat = uuidSat;  
 }  
  
  
 @Column(name = "DIAS\_CREDITO", precision = 22, scale = 0)  
 public BigDecimal getDiasCredito() {  
 return this.diasCredito;  
 }  
  
 public void setDiasCredito(BigDecimal diasCredito) {  
 this.diasCredito = diasCredito;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name = "FECHA\_REVISION", length = 7)  
 public Date getFechaRevision() {  
 return this.fechaRevision;  
 }  
  
 public void setFechaRevision(Date fechaRevision) {  
 this.fechaRevision = fechaRevision;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name = "FECHA\_VENCIMIENTO", length = 7)  
 public Date getFechaVencimiento() {  
 return this.fechaVencimiento;  
 }  
  
 public void setFechaVencimiento(Date fechaVencimiento) {  
 this.fechaVencimiento = fechaVencimiento;  
 }  
  
  
 @Column(name = "CUENTA")  
 public String getCuenta() {  
 return this.cuenta;  
 }  
  
 public void setCuenta(String cuenta) {  
 this.cuenta = cuenta;  
 }  
  
  
 @Column(name = "CONFIRMACION\_SAT")  
 public String getConfirmacionSat() {  
 return this.confirmacionSat;  
 }  
  
 public void setConfirmacionSat(String confirmacionSat) {  
 this.confirmacionSat = confirmacionSat;  
 }  
  
  
 @Column(name = "CONDICIONES\_PAGO")  
 public String getCondicionesPago() {  
 return this.condicionesPago;  
 }  
  
 public void setCondicionesPago(String condicionesPago) {  
 this.condicionesPago = condicionesPago;  
 }  
  
  
 @Column(name = "SELLO\_DIGITAL\_EMISOR")  
 public String getSelloDigitalEmisor() {  
 return this.selloDigitalEmisor;  
 }  
  
 public void setSelloDigitalEmisor(String selloDigitalEmisor) {  
 this.selloDigitalEmisor = selloDigitalEmisor;  
 }  
  
  
 @Column(name = "SELLO\_DIGITA\_SAT")  
 public String getSelloDigitaSat() {  
 return this.selloDigitaSat;  
 }  
  
 public void setSelloDigitaSat(String selloDigitaSat) {  
 this.selloDigitaSat = selloDigitaSat;  
 }  
  
  
 @Column(name = "CADENA\_ORIGINAL")  
 public String getCadenaOriginal() {  
 return this.cadenaOriginal;  
 }  
  
 public void setCadenaOriginal(String cadenaOriginal) {  
 this.cadenaOriginal = cadenaOriginal;  
 }  
  
  
 @Column(name = "TIMBRADO", length = 50)  
 public String getTimbrado() {  
 return this.timbrado;  
 }  
  
 public void setTimbrado(String timbrado) {  
 this.timbrado = timbrado;  
 }  
  
  
 @Column(name = "CERTIFICADO")  
 public String getCertificado() {  
 return this.certificado;  
 }  
  
 public void setCertificado(String certificado) {  
 this.certificado = certificado;  
 }  
  
  
 @Column(name = "NO\_CERTIFICADO", length = 15)  
 public String getNoCertificado() {  
 return this.noCertificado;  
 }  
  
 public void setNoCertificado(String noCertificado) {  
 this.noCertificado = noCertificado;  
 }  
  
 @Column(name = "NO\_CERTIFICADO\_SAT", length = 40)  
 public String getNoCertificadoSat() {  
 return this.noCertificadoSat;  
 }  
  
 public void setNoCertificadoSat(String noCertificadoSat) {  
 this.noCertificadoSat = noCertificadoSat;  
 }  
  
 @Column(name = "STR\_XML", length = 4000)  
 public String getStrXml() {  
 return this.strXml;  
 }  
  
 public void setStrXml(String strXml) {  
 this.strXml = strXml;  
 }  
  
  
 @Column(name = "STR\_XML\_CANCELACION", length = 4000)  
 public String getStrXmlCancelacion() {  
 return this.strXmlCancelacion;  
 }  
  
 public void setStrXmlCancelacion(String strXmlCancelacion) {  
 this.strXmlCancelacion = strXmlCancelacion;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name = "FECHA\_HORA\_CANCELACION", length = 11)  
 public Date getFechaHoraCancelacion() {  
 return this.fechaHoraCancelacion;  
 }  
  
 public void setFechaHoraCancelacion(Date fechaHoraCancelacion) {  
 this.fechaHoraCancelacion = fechaHoraCancelacion;  
 }  
  
  
 @Column(name = "OBSERVACIONES")  
 public String getObservaciones() {  
 return this.observaciones;  
 }  
  
 public void setObservaciones(String observaciones) {  
 this.observaciones = observaciones;  
 }  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "txDocumentosByDocumentoId")  
 public Set<TxFacturaUuids> getTxFacturaUuidsesForDocumentoId() {  
 return this.txFacturaUuidsesForDocumentoId;  
 }  
  
 public void setTxFacturaUuidsesForDocumentoId(Set<TxFacturaUuids> txFacturaUuidsesForDocumentoId) {  
 this.txFacturaUuidsesForDocumentoId = txFacturaUuidsesForDocumentoId;  
 }  
  
 @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.EAGER, mappedBy = "txDocumentos")  
 public Set<CatFacturaProductos> getCatFacturaProductoses() {  
 return this.catFacturaProductoses;  
 }  
  
 public void setCatFacturaProductoses(Set<CatFacturaProductos> catFacturaProductoses) {  
 this.catFacturaProductoses = catFacturaProductoses;  
 }  
  
 @OneToMany(fetch = FetchType.LAZY, mappedBy = "txDocumentosByDocumentoAsociadoId")  
 public Set<TxFacturaUuids> getTxFacturaUuidsesForDocumentoAsociadoId() {  
 return this.txFacturaUuidsesForDocumentoAsociadoId;  
 }  
  
 public void setTxFacturaUuidsesForDocumentoAsociadoId(Set<TxFacturaUuids> txFacturaUuidsesForDocumentoAsociadoId) {  
 this.txFacturaUuidsesForDocumentoAsociadoId = txFacturaUuidsesForDocumentoAsociadoId;  
 }  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxErrorCargaCrudos.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.Date;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* TxErrorCargaCrudos generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_ERROR\_CARGA\_CRUDOS"  
)  
public class TxErrorCargaCrudos implements java.io.Serializable {  
  
  
 private BigDecimal errorCargaCrudosId;  
 private BigDecimal tipoError;  
 private String errm;  
 private String data;  
 private Date errorDate;  
  
 public TxErrorCargaCrudos() {  
 }  
  
 public TxErrorCargaCrudos(BigDecimal tipoError, String errm, String data, Date errorDate) {  
 this.tipoError = tipoError;  
 this.errm = errm;  
 this.data = data;  
 this.errorDate = errorDate;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="ERROR\_CARGA\_CRUDOS\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getErrorCargaCrudosId() {  
 return this.errorCargaCrudosId;  
 }  
   
 public void setErrorCargaCrudosId(BigDecimal errorCargaCrudosId) {  
 this.errorCargaCrudosId = errorCargaCrudosId;  
 }  
  
   
 @Column(name="TIPO\_ERROR", precision=22, scale=0)  
 public BigDecimal getTipoError() {  
 return this.tipoError;  
 }  
   
 public void setTipoError(BigDecimal tipoError) {  
 this.tipoError = tipoError;  
 }  
  
   
 @Column(name="ERRM", length=4000)  
 public String getErrm() {  
 return this.errm;  
 }  
   
 public void setErrm(String errm) {  
 this.errm = errm;  
 }  
  
   
 @Column(name="DATA", length=4000)  
 public String getData() {  
 return this.data;  
 }  
   
 public void setData(String data) {  
 this.data = data;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="ERROR\_DATE", length=11)  
 public Date getErrorDate() {  
 return this.errorDate;  
 }  
   
 public void setErrorDate(Date errorDate) {  
 this.errorDate = errorDate;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxErrorFormato.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.Date;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* TxErrorFormato generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_ERROR\_FORMATO"  
)  
public class TxErrorFormato implements java.io.Serializable {  
  
  
 private BigDecimal errorFormatoId;  
 private BigDecimal crudosId;  
 private String valor;  
 private String columnaCincominutal;  
 private Date fecha;  
 private String error;  
  
 public TxErrorFormato() {  
 }  
  
   
 public TxErrorFormato(Date fecha) {  
 this.fecha = fecha;  
 }  
 public TxErrorFormato(BigDecimal crudosId, String valor, String columnaCincominutal, Date fecha, String error) {  
 this.crudosId = crudosId;  
 this.valor = valor;  
 this.columnaCincominutal = columnaCincominutal;  
 this.fecha = fecha;  
 this.error = error;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="ERROR\_FORMATO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getErrorFormatoId() {  
 return this.errorFormatoId;  
 }  
   
 public void setErrorFormatoId(BigDecimal errorFormatoId) {  
 this.errorFormatoId = errorFormatoId;  
 }  
  
   
 @Column(name="CRUDOS\_ID", precision=22, scale=0)  
 public BigDecimal getCrudosId() {  
 return this.crudosId;  
 }  
   
 public void setCrudosId(BigDecimal crudosId) {  
 this.crudosId = crudosId;  
 }  
  
   
 @Column(name="VALOR", length=200)  
 public String getValor() {  
 return this.valor;  
 }  
   
 public void setValor(String valor) {  
 this.valor = valor;  
 }  
  
   
 @Column(name="COLUMNA\_CINCOMINUTAL", length=200)  
 public String getColumnaCincominutal() {  
 return this.columnaCincominutal;  
 }  
   
 public void setColumnaCincominutal(String columnaCincominutal) {  
 this.columnaCincominutal = columnaCincominutal;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="FECHA", nullable=false, length=7)  
 public Date getFecha() {  
 return this.fecha;  
 }  
   
 public void setFecha(Date fecha) {  
 this.fecha = fecha;  
 }  
  
   
 @Column(name="ERROR", length=2000)  
 public String getError() {  
 return this.error;  
 }  
   
 public void setError(String error) {  
 this.error = error;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxEscenario.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* TxEscenario generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_ESCENARIO"  
)  
public class TxEscenario implements java.io.Serializable {  
  
  
 private BigDecimal escenarioId;  
 private CatEtapas catEtapas;  
 private BigDecimal principal;  
 private String nombre;  
 private Set<TxEscenarioVariable> txEscenarioVariables = new HashSet<TxEscenarioVariable>(0);  
 private Set<TxFormulas> txFormulases = new HashSet<TxFormulas>(0);  
 private Set<TxDocumentos> txDocumentoses = new HashSet<TxDocumentos>(0);  
  
 public TxEscenario() {  
 }  
  
 public TxEscenario(CatEtapas catEtapas, BigDecimal principal, String nombre, Set<TxEscenarioVariable> txEscenarioVariables, Set<TxFormulas> txFormulases, Set<TxDocumentos> txDocumentoses) {  
 this.catEtapas = catEtapas;  
 this.principal = principal;  
 this.nombre = nombre;  
 this.txEscenarioVariables = txEscenarioVariables;  
 this.txFormulases = txFormulases;  
 this.txDocumentoses = txDocumentoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="ESCENARIO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getEscenarioId() {  
 return this.escenarioId;  
 }  
   
 public void setEscenarioId(BigDecimal escenarioId) {  
 this.escenarioId = escenarioId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="ETAPA\_ID")  
 public CatEtapas getCatEtapas() {  
 return this.catEtapas;  
 }  
   
 public void setCatEtapas(CatEtapas catEtapas) {  
 this.catEtapas = catEtapas;  
 }  
  
   
 @Column(name="PRINCIPAL", precision=22, scale=0)  
 public BigDecimal getPrincipal() {  
 return this.principal;  
 }  
   
 public void setPrincipal(BigDecimal principal) {  
 this.principal = principal;  
 }  
  
   
 @Column(name="NOMBRE", length=200)  
 public String getNombre() {  
 return this.nombre;  
 }  
   
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="txEscenario")  
 public Set<TxEscenarioVariable> getTxEscenarioVariables() {  
 return this.txEscenarioVariables;  
 }  
   
 public void setTxEscenarioVariables(Set<TxEscenarioVariable> txEscenarioVariables) {  
 this.txEscenarioVariables = txEscenarioVariables;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="txEscenario")  
 public Set<TxFormulas> getTxFormulases() {  
 return this.txFormulases;  
 }  
   
 public void setTxFormulases(Set<TxFormulas> txFormulases) {  
 this.txFormulases = txFormulases;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="txEscenario")  
 public Set<TxDocumentos> getTxDocumentoses() {  
 return this.txDocumentoses;  
 }  
   
 public void setTxDocumentoses(Set<TxDocumentos> txDocumentoses) {  
 this.txDocumentoses = txDocumentoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxEscenariosBitacora.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.Date;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* TxEscenariosBitacora generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_ESCENARIOS\_BITACORA"  
)  
public class TxEscenariosBitacora implements java.io.Serializable {  
  
  
 private BigDecimal escenarioBitacoraId;  
 private TxEvento txEvento;  
 private String escenario;  
 private Date fechaInicio;  
 private Date fechaFinal;  
 private String informes;  
  
 public TxEscenariosBitacora() {  
 }  
  
 public TxEscenariosBitacora(TxEvento txEvento, String escenario, Date fechaInicio, Date fechaFinal, String informes) {  
 this.txEvento = txEvento;  
 this.escenario = escenario;  
 this.fechaInicio = fechaInicio;  
 this.fechaFinal = fechaFinal;  
 this.informes = informes;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="ESCENARIO\_BITACORA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getEscenarioBitacoraId() {  
 return this.escenarioBitacoraId;  
 }  
   
 public void setEscenarioBitacoraId(BigDecimal escenarioBitacoraId) {  
 this.escenarioBitacoraId = escenarioBitacoraId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="EVENTO\_ID", nullable=false)  
 public TxEvento getTxEvento() {  
 return this.txEvento;  
 }  
   
 public void setTxEvento(TxEvento txEvento) {  
 this.txEvento = txEvento;  
 }  
  
   
 @Column(name="ESCENARIO", nullable=false, length=100)  
 public String getEscenario() {  
 return this.escenario;  
 }  
   
 public void setEscenario(String escenario) {  
 this.escenario = escenario;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="FECHA\_INICIO", nullable=false, length=11)  
 public Date getFechaInicio() {  
 return this.fechaInicio;  
 }  
   
 public void setFechaInicio(Date fechaInicio) {  
 this.fechaInicio = fechaInicio;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="FECHA\_FINAL", nullable=false, length=11)  
 public Date getFechaFinal() {  
 return this.fechaFinal;  
 }  
   
 public void setFechaFinal(Date fechaFinal) {  
 this.fechaFinal = fechaFinal;  
 }  
  
   
 @Column(name="INFORMES", nullable=false, length=1500)  
 public String getInformes() {  
 return this.informes;  
 }  
   
 public void setInformes(String informes) {  
 this.informes = informes;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxEscenarioVariable.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.Date;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
import javax.persistence.UniqueConstraint;  
  
/\*\*  
 \* TxEscenarioVariable generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_ESCENARIO\_VARIABLE"  
 , uniqueConstraints = @UniqueConstraint(columnNames="CINCOMINUTAL\_VAR\_ID")   
)  
public class TxEscenarioVariable implements java.io.Serializable {  
  
  
 private BigDecimal escenarioVariableId;  
 private TxEscenario txEscenario;  
 private BigDecimal cincominutalVarId;  
 private BigDecimal principal;  
 private BigDecimal numEscenario;  
 private BigDecimal valor;  
 private Date fecha;  
 private BigDecimal factorCorrecion;  
  
 public TxEscenarioVariable() {  
 }  
  
 public TxEscenarioVariable(TxEscenario txEscenario, BigDecimal cincominutalVarId, BigDecimal principal, BigDecimal numEscenario, BigDecimal valor, Date fecha, BigDecimal factorCorrecion) {  
 this.txEscenario = txEscenario;  
 this.cincominutalVarId = cincominutalVarId;  
 this.principal = principal;  
 this.numEscenario = numEscenario;  
 this.valor = valor;  
 this.fecha = fecha;  
 this.factorCorrecion = factorCorrecion;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="ESCENARIO\_VARIABLE\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getEscenarioVariableId() {  
 return this.escenarioVariableId;  
 }  
   
 public void setEscenarioVariableId(BigDecimal escenarioVariableId) {  
 this.escenarioVariableId = escenarioVariableId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="ESCENARIO\_ID")  
 public TxEscenario getTxEscenario() {  
 return this.txEscenario;  
 }  
   
 public void setTxEscenario(TxEscenario txEscenario) {  
 this.txEscenario = txEscenario;  
 }  
  
   
 @Column(name="CINCOMINUTAL\_VAR\_ID", unique=true, precision=22, scale=0)  
 public BigDecimal getCincominutalVarId() {  
 return this.cincominutalVarId;  
 }  
   
 public void setCincominutalVarId(BigDecimal cincominutalVarId) {  
 this.cincominutalVarId = cincominutalVarId;  
 }  
  
   
 @Column(name="PRINCIPAL", precision=22, scale=0)  
 public BigDecimal getPrincipal() {  
 return this.principal;  
 }  
   
 public void setPrincipal(BigDecimal principal) {  
 this.principal = principal;  
 }  
  
   
 @Column(name="NUM\_ESCENARIO", precision=22, scale=0)  
 public BigDecimal getNumEscenario() {  
 return this.numEscenario;  
 }  
   
 public void setNumEscenario(BigDecimal numEscenario) {  
 this.numEscenario = numEscenario;  
 }  
  
   
 @Column(name="VALOR", precision=22, scale=0)  
 public BigDecimal getValor() {  
 return this.valor;  
 }  
   
 public void setValor(BigDecimal valor) {  
 this.valor = valor;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="FECHA", length=11)  
 public Date getFecha() {  
 return this.fecha;  
 }  
   
 public void setFecha(Date fecha) {  
 this.fecha = fecha;  
 }  
  
   
 @Column(name="FACTOR\_CORRECION", precision=22, scale=0)  
 public BigDecimal getFactorCorrecion() {  
 return this.factorCorrecion;  
 }  
   
 public void setFactorCorrecion(BigDecimal factorCorrecion) {  
 this.factorCorrecion = factorCorrecion;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxEvento.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
  
import java.math.BigDecimal;  
import java.math.BigInteger;  
  
import java.util.Date;  
import java.util.Set;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
import javax.validation.constraints.Size;  
  
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlTransient;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "TX\_EVENTO")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "TxEvento.findAll", query = "SELECT t FROM TxEvento t")  
 , @NamedQuery(name = "TxEvento.findByEventoId", query = "SELECT t FROM TxEvento t WHERE t.eventoId = :eventoId")  
 , @NamedQuery(name = "TxEvento.findByFechaInicio", query = "SELECT t FROM TxEvento t WHERE t.fechaInicio = :fechaInicio")  
 , @NamedQuery(name = "TxEvento.findByFechaFinal", query = "SELECT t FROM TxEvento t WHERE t.fechaFinal = :fechaFinal")  
 , @NamedQuery(name = "TxEvento.findByInformes", query = "SELECT t FROM TxEvento t WHERE t.informes = :informes")  
 , @NamedQuery(name = "TxEvento.findByNombre", query = "SELECT t FROM TxEvento t WHERE t.nombre = :nombre")  
 , @NamedQuery(name = "TxEvento.findByDescripcion", query = "SELECT t FROM TxEvento t WHERE t.descripcion = :descripcion")  
 , @NamedQuery(name = "TxEvento.findByPotencia", query = "SELECT t FROM TxEvento t WHERE t.potencia = :potencia")  
 , @NamedQuery(name = "TxEvento.findByFalla", query = "SELECT t FROM TxEvento t WHERE t.falla = :falla")  
 , @NamedQuery(name = "TxEvento.findByLicencia", query = "SELECT t FROM TxEvento t WHERE t.licencia = :licencia")  
 , @NamedQuery(name = "TxEvento.findByConcepto", query = "SELECT t FROM TxEvento t WHERE t.concepto = :concepto")  
 , @NamedQuery(name = "TxEvento.findByObservaciones", query = "SELECT t FROM TxEvento t WHERE t.observaciones = :observaciones")})  
public class TxEvento implements Serializable {  
  
 @OneToMany(mappedBy = "eventoId")  
 private Set<TxAdjuntos> txAdjuntosSet;  
  
 @Column(name = "PERDIDA\_POTENCIA")  
 private BigInteger perdidaPotencia;  
 @Size(max = 1020)  
 @Column(name = "OT")  
 private String ot;  
 @Size(max = 1020)  
 @Column(name = "MW\_OFERTADOS")  
 private String mwOfertados;  
 @Size(max = 1020)  
 @Column(name = "CARGA\_INICIAL\_MW")  
 private String cargaInicialMw;  
 @Size(max = 1020)  
 @Column(name = "CARGA\_FINAL\_MW")  
 private String cargaFinalMw;  
 @Size(max = 100)  
 @Column(name = "NOMBRE\_OPERADOR\_CENACE\_ABRIO")  
 private String nombreOperadorCenaceAbrio;  
 @Size(max = 100)  
 @Column(name = "APELLIDOS\_OPERADOR\_CENACE\_ABRIO")  
 private String apellidosOperadorCenaceAbrio;  
 @Size(max = 100)  
 @Column(name = "NOMBRE\_OPERADOR\_CENACE\_CERRO")  
 private String nombreOperadorCenaceCerro;  
 @Size(max = 100)  
 @Column(name = "APELLIDOS\_OPERADOR\_CENACE\_CERRO")  
 private String apellidosOperadorCenaceCerro;  
 @Size(max = 100)  
 @Column(name = "NOMBRE\_OPERADOR\_PLANTA\_ABRIO")  
 private String nombreOperadorPlantaAbrio;  
 @Size(max = 100)  
 @Column(name = "NOMBRE\_OPERADOR\_PLANTA\_CERRO")  
 private String nombreOperadorPlantaCerro;  
  
 @OneToMany(mappedBy = "eventoId")  
 private Set<TxEventoDetalles> txEventoDetallesSet;  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id @GeneratedValue(strategy=IDENTITY)  
 @Column(name = "EVENTO\_ID")  
 private BigDecimal eventoId;  
 @Column(name = "FECHA\_INICIO")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaInicio;  
 @Column(name = "FECHA\_INICIO\_REDONDEO")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaInicioRedondeo;  
 @Column(name = "FECHA\_FINAL")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaFinal;  
 @Column(name = "FECHA\_FINAL\_REDONDEO")  
 @Temporal(TemporalType.TIMESTAMP)  
 private Date fechaFinalRedondeo;  
 @Size(min = 1, max = 1500)  
 @Column(name = "INFORMES")  
 private String informes;  
 @Size(max = 1020)  
 @Column(name = "NOMBRE")  
 private String nombre;  
 @Size(max = 1020)  
 @Column(name = "DESCRIPCION")  
 private String descripcion;  
 @Column(name = "POTENCIA")  
 private BigInteger potencia;  
 @Size(max = 120)  
 @Column(name = "FALLA")  
 private String falla;  
 @Size(max = 1020)  
 @Column(name = "LICENCIA")  
 private String licencia;  
 @Size(max = 1020)  
 @Column(name = "CONCEPTO")  
 private String concepto;  
 @Size(max = 1020)  
 @Column(name = "OBSERVACIONES")  
 private String observaciones;   
 @JoinColumn(name = "CLASIFICACION\_EVENTO\_ID", referencedColumnName = "CLASIFICACION\_EVENTO\_ID")  
 @ManyToOne  
 private CatClasificacionEvento clasificacionEventoId;  
 @JoinColumn(name = "BITACORA\_ID", referencedColumnName = "BITACORA\_ID")  
 @ManyToOne  
 private TxBitacora bitacoraId;  
  
 @JoinColumn(name = "USUARIO\_ID", referencedColumnName = "USUARIO\_ID")  
 @ManyToOne  
 private TxUsuario usuarioId;  
  
  
 public void setUsuarioId(TxUsuario usuarioId) {  
 this.usuarioId = usuarioId;  
 }  
  
 public TxUsuario getUsuarioId() {  
 return usuarioId;  
 }  
  
 public TxEvento() {  
 }  
  
 public void setFechaInicioRedondeo(Date fechaInicioRedondeo) {  
 this.fechaInicioRedondeo = fechaInicioRedondeo;  
 }  
  
 public Date getFechaInicioRedondeo() {  
 return fechaInicioRedondeo;  
 }  
  
 public void setFechaFinalRedondeo(Date fechaFinalRedondeo) {  
 this.fechaFinalRedondeo = fechaFinalRedondeo;  
 }  
  
 public Date getFechaFinalRedondeo() {  
 return fechaFinalRedondeo;  
 }  
  
 public TxEvento(BigDecimal eventoId) {  
 this.eventoId = eventoId;  
 }  
  
 public TxEvento(BigDecimal eventoId, Date fechaInicio, Date fechaFinal, String informes, short column1) {  
 this.eventoId = eventoId;  
 this.fechaInicio = fechaInicio;  
 this.fechaFinal = fechaFinal;  
 this.informes = informes;  
 }  
  
 public BigDecimal getEventoId() {  
 return eventoId;  
 }  
  
 public void setEventoId(BigDecimal eventoId) {  
 this.eventoId = eventoId;  
 }  
  
 public Date getFechaInicio() {  
 return fechaInicio;  
 }  
  
 public void setFechaInicio(Date fechaInicio) {  
 this.fechaInicio = fechaInicio;  
 }  
  
 public Date getFechaFinal() {  
 return fechaFinal;  
 }  
  
 public void setFechaFinal(Date fechaFinal) {  
 this.fechaFinal = fechaFinal;  
 }  
  
 public String getInformes() {  
 return informes;  
 }  
  
 public void setInformes(String informes) {  
 this.informes = informes;  
 }  
  
 public String getNombre() {  
 return nombre;  
 }  
  
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
 public String getDescripcion() {  
 return descripcion;  
 }  
  
 public void setDescripcion(String descripcion) {  
 this.descripcion = descripcion;  
 }  
  
 public BigInteger getPotencia() {  
 return potencia;  
 }  
  
 public void setPotencia(BigInteger potencia) {  
 this.potencia = potencia;  
 }  
  
 public String getFalla() {  
 return falla;  
 }  
  
 public void setFalla(String falla) {  
 this.falla = falla;  
 }  
  
 public String getLicencia() {  
 return licencia;  
 }  
  
 public void setLicencia(String licencia) {  
 this.licencia = licencia;  
 }  
  
 public String getConcepto() {  
 return concepto;  
 }  
  
 public void setConcepto(String concepto) {  
 this.concepto = concepto;  
 }  
  
 public String getObservaciones() {  
 return observaciones;  
 }  
  
 public void setObservaciones(String observaciones) {  
 this.observaciones = observaciones;  
 }  
  
  
 public CatClasificacionEvento getClasificacionEventoId() {  
 return clasificacionEventoId;  
 }  
  
 public void setClasificacionEventoId(CatClasificacionEvento clasificacionEventoId) {  
 this.clasificacionEventoId = clasificacionEventoId;  
 }  
  
 public TxBitacora getBitacoraId() {  
 return bitacoraId;  
 }  
  
 public void setBitacoraId(TxBitacora bitacoraId) {  
 this.bitacoraId = bitacoraId;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (eventoId != null ? eventoId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof TxEvento)) {  
 return false;  
 }  
 TxEvento other = (TxEvento) object;  
 if ((this.eventoId == null && other.eventoId != null) || (this.eventoId != null && !this.eventoId.equals(other.eventoId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.TxEvento[ eventoId=" + eventoId + " ]";  
 }  
  
 @XmlTransient  
 public Set<TxEventoDetalles> getTxEventoDetallesSet() {  
 return txEventoDetallesSet;  
 }  
  
 public void setTxEventoDetallesSet(Set<TxEventoDetalles> txEventoDetallesSet) {  
 this.txEventoDetallesSet = txEventoDetallesSet;  
 }  
  
 public BigInteger getPerdidaPotencia() {  
 return perdidaPotencia;  
 }  
  
 public void setPerdidaPotencia(BigInteger perdidaPotencia) {  
 this.perdidaPotencia = perdidaPotencia;  
 }  
  
 public String getOt() {  
 return ot;  
 }  
  
 public void setOt(String ot) {  
 this.ot = ot;  
 }  
  
 public String getMwOfertados() {  
 return mwOfertados;  
 }  
  
 public void setMwOfertados(String mwOfertados) {  
 this.mwOfertados = mwOfertados;  
 }  
  
 public String getCargaInicialMw() {  
 return cargaInicialMw;  
 }  
  
 public void setCargaInicialMw(String cargaInicialMw) {  
 this.cargaInicialMw = cargaInicialMw;  
 }  
  
 public String getCargaFinalMw() {  
 return cargaFinalMw;  
 }  
  
 public void setCargaFinalMw(String cargaFinalMw) {  
 this.cargaFinalMw = cargaFinalMw;  
 }  
  
 public String getNombreOperadorCenaceAbrio() {  
 return nombreOperadorCenaceAbrio;  
 }  
  
 public void setNombreOperadorCenaceAbrio(String nombreOperadorCenaceAbrio) {  
 this.nombreOperadorCenaceAbrio = nombreOperadorCenaceAbrio;  
 }  
  
 public String getApellidosOperadorCenaceAbrio() {  
 return apellidosOperadorCenaceAbrio;  
 }  
  
 public void setApellidosOperadorCenaceAbrio(String apellidosOperadorCenaceAbrio) {  
 this.apellidosOperadorCenaceAbrio = apellidosOperadorCenaceAbrio;  
 }  
  
 public String getNombreOperadorCenaceCerro() {  
 return nombreOperadorCenaceCerro;  
 }  
  
 public void setNombreOperadorCenaceCerro(String nombreOperadorCenaceCerro) {  
 this.nombreOperadorCenaceCerro = nombreOperadorCenaceCerro;  
 }  
  
 public String getApellidosOperadorCenaceCerro() {  
 return apellidosOperadorCenaceCerro;  
 }  
  
 public void setApellidosOperadorCenaceCerro(String apellidosOperadorCenaceCerro) {  
 this.apellidosOperadorCenaceCerro = apellidosOperadorCenaceCerro;  
 }  
  
 public String getNombreOperadorPlantaAbrio() {  
 return nombreOperadorPlantaAbrio;  
 }  
  
 public void setNombreOperadorPlantaAbrio(String nombreOperadorPlantaAbrio) {  
 this.nombreOperadorPlantaAbrio = nombreOperadorPlantaAbrio;  
 }  
  
 public String getNombreOperadorPlantaCerro() {  
 return nombreOperadorPlantaCerro;  
 }  
  
 public void setNombreOperadorPlantaCerro(String nombreOperadorPlantaCerro) {  
 this.nombreOperadorPlantaCerro = nombreOperadorPlantaCerro;  
 }  
  
 @XmlTransient  
 public Set<TxAdjuntos> getTxAdjuntosSet() {  
 return txAdjuntosSet;  
 }  
  
 public void setTxAdjuntosSet(Set<TxAdjuntos> txAdjuntosSet) {  
 this.txAdjuntosSet = txAdjuntosSet;  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxEventoDetalles.java

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package mx.mbi.dgc.jpa.entity;  
  
import java.io.Serializable;  
  
import java.math.BigDecimal;  
  
import javax.persistence.Basic;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.NamedQueries;  
import javax.persistence.NamedQuery;  
import javax.persistence.Table;  
  
import javax.xml.bind.annotation.XmlRootElement;  
  
/\*\*  
 \*  
 \* @author francisco.valdez  
 \*/  
@Entity  
@Table(name = "TX\_EVENTO\_DETALLES")  
@XmlRootElement  
@NamedQueries({  
 @NamedQuery(name = "TxEventoDetalles.findAll", query = "SELECT t FROM TxEventoDetalles t")  
 , @NamedQuery(name = "TxEventoDetalles.findByEventoDetalleId", query = "SELECT t FROM TxEventoDetalles t WHERE t.eventoDetalleId = :eventoDetalleId")})  
public class TxEventoDetalles implements Serializable {  
  
 private static final long serialVersionUID = 1L;  
 // @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to enforce field validation  
 @Id @GeneratedValue(strategy=IDENTITY)  
 @Basic(optional = false)  
 @Column(name = "EVENTO\_DETALLE\_ID")  
 private BigDecimal eventoDetalleId;  
 @JoinColumn(name = "CONFIGURACION\_OPCIONES\_ID", referencedColumnName = "CONFIGURACION\_OPCIONES\_ID")  
 @ManyToOne  
 private CatConfiguracionOpciones configuracionOpcionesId;  
 @JoinColumn(name = "EVENTO\_ID", referencedColumnName = "EVENTO\_ID")  
 @ManyToOne  
 private TxEvento eventoId;  
  
 public TxEventoDetalles() {  
 }  
  
 public TxEventoDetalles(BigDecimal eventoDetalleId) {  
 this.eventoDetalleId = eventoDetalleId;  
 }  
  
 public BigDecimal getEventoDetalleId() {  
 return eventoDetalleId;  
 }  
  
 public void setEventoDetalleId(BigDecimal eventoDetalleId) {  
 this.eventoDetalleId = eventoDetalleId;  
 }  
  
 public CatConfiguracionOpciones getConfiguracionOpcionesId() {  
 return configuracionOpcionesId;  
 }  
  
 public void setConfiguracionOpcionesId(CatConfiguracionOpciones configuracionOpcionesId) {  
 this.configuracionOpcionesId = configuracionOpcionesId;  
 }  
  
 public TxEvento getEventoId() {  
 return eventoId;  
 }  
  
 public void setEventoId(TxEvento eventoId) {  
 this.eventoId = eventoId;  
 }  
  
 @Override  
 public int hashCode() {  
 int hash = 0;  
 hash += (eventoDetalleId != null ? eventoDetalleId.hashCode() : 0);  
 return hash;  
 }  
  
 @Override  
 public boolean equals(Object object) {  
 // TODO: Warning - this method won't work in the case the id fields are not set  
 if (!(object instanceof TxEventoDetalles)) {  
 return false;  
 }  
 TxEventoDetalles other = (TxEventoDetalles) object;  
 if ((this.eventoDetalleId == null && other.eventoDetalleId != null) || (this.eventoDetalleId != null && !this.eventoDetalleId.equals(other.eventoDetalleId))) {  
 return false;  
 }  
 return true;  
 }  
  
 @Override  
 public String toString() {  
 return "mx.mbi.dgc.jpa.entity.TxEventoDetalles[ eventoDetalleId=" + eventoDetalleId + " ]";  
 }  
   
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxFacturaUuids.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* TxFacturaUuids generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_FACTURA\_UUIDS"  
)  
public class TxFacturaUuids implements java.io.Serializable {  
  
  
 private BigDecimal facturaUuidId;  
 private TxDocumentos txDocumentosByDocumentoId;  
 private TxDocumentos txDocumentosByDocumentoAsociadoId;  
 private CatTipoRelacion catTipoRelacion;  
 private BigDecimal tipoDocumentoId;  
 private String observaciones;  
  
 public TxFacturaUuids() {  
 }  
  
 public TxFacturaUuids(TxDocumentos txDocumentosByDocumentoId, TxDocumentos txDocumentosByDocumentoAsociadoId, CatTipoRelacion catTipoRelacion, BigDecimal tipoDocumentoId, String observaciones) {  
 this.txDocumentosByDocumentoId = txDocumentosByDocumentoId;  
 this.txDocumentosByDocumentoAsociadoId = txDocumentosByDocumentoAsociadoId;  
 this.catTipoRelacion = catTipoRelacion;  
 this.tipoDocumentoId = tipoDocumentoId;  
 this.observaciones = observaciones;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="FACTURA\_UUID\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getFacturaUuidId() {  
 return this.facturaUuidId;  
 }  
   
 public void setFacturaUuidId(BigDecimal facturaUuidId) {  
 this.facturaUuidId = facturaUuidId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="DOCUMENTO\_ID")  
 public TxDocumentos getTxDocumentosByDocumentoId() {  
 return this.txDocumentosByDocumentoId;  
 }  
   
 public void setTxDocumentosByDocumentoId(TxDocumentos txDocumentosByDocumentoId) {  
 this.txDocumentosByDocumentoId = txDocumentosByDocumentoId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="DOCUMENTO\_ASOCIADO\_ID")  
 public TxDocumentos getTxDocumentosByDocumentoAsociadoId() {  
 return this.txDocumentosByDocumentoAsociadoId;  
 }  
   
 public void setTxDocumentosByDocumentoAsociadoId(TxDocumentos txDocumentosByDocumentoAsociadoId) {  
 this.txDocumentosByDocumentoAsociadoId = txDocumentosByDocumentoAsociadoId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="TIPO\_RELACION\_ID")  
 public CatTipoRelacion getCatTipoRelacion() {  
 return this.catTipoRelacion;  
 }  
   
 public void setCatTipoRelacion(CatTipoRelacion catTipoRelacion) {  
 this.catTipoRelacion = catTipoRelacion;  
 }  
  
   
 @Column(name="TIPO\_DOCUMENTO\_ID", precision=22, scale=0)  
 public BigDecimal getTipoDocumentoId() {  
 return this.tipoDocumentoId;  
 }  
   
 public void setTipoDocumentoId(BigDecimal tipoDocumentoId) {  
 this.tipoDocumentoId = tipoDocumentoId;  
 }  
  
   
 @Column(name="OBSERVACIONES")  
 public String getObservaciones() {  
 return this.observaciones;  
 }  
   
 public void setObservaciones(String observaciones) {  
 this.observaciones = observaciones;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxFormulas.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
  
/\*\*  
 \* TxFormulas generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_FORMULAS"  
)  
public class TxFormulas implements java.io.Serializable {  
  
  
 private BigDecimal formulaId;  
 private CatAnioMes catAnioMes;  
 private TxEscenario txEscenario;  
 private String formula;  
 private BigDecimal valor;  
 private Set<CatFormulaComponentes> catFormulaComponenteses = new HashSet<CatFormulaComponentes>(0);  
  
 public TxFormulas() {  
 }  
  
 public TxFormulas(CatAnioMes catAnioMes, TxEscenario txEscenario, String formula, BigDecimal valor, Set<CatFormulaComponentes> catFormulaComponenteses) {  
 this.catAnioMes = catAnioMes;  
 this.txEscenario = txEscenario;  
 this.formula = formula;  
 this.valor = valor;  
 this.catFormulaComponenteses = catFormulaComponenteses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="FORMULA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getFormulaId() {  
 return this.formulaId;  
 }  
   
 public void setFormulaId(BigDecimal formulaId) {  
 this.formulaId = formulaId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="ANIO\_MES\_ID")  
 public CatAnioMes getCatAnioMes() {  
 return this.catAnioMes;  
 }  
   
 public void setCatAnioMes(CatAnioMes catAnioMes) {  
 this.catAnioMes = catAnioMes;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="ESCENARIO\_ID")  
 public TxEscenario getTxEscenario() {  
 return this.txEscenario;  
 }  
   
 public void setTxEscenario(TxEscenario txEscenario) {  
 this.txEscenario = txEscenario;  
 }  
  
   
 @Column(name="FORMULA", length=50)  
 public String getFormula() {  
 return this.formula;  
 }  
   
 public void setFormula(String formula) {  
 this.formula = formula;  
 }  
  
   
 @Column(name="VALOR", precision=22, scale=0)  
 public BigDecimal getValor() {  
 return this.valor;  
 }  
   
 public void setValor(BigDecimal valor) {  
 this.valor = valor;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="txFormulas")  
 public Set<CatFormulaComponentes> getCatFormulaComponenteses() {  
 return this.catFormulaComponenteses;  
 }  
   
 public void setCatFormulaComponenteses(Set<CatFormulaComponentes> catFormulaComponenteses) {  
 this.catFormulaComponenteses = catFormulaComponenteses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxLogErrorCarga.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.Date;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* TxLogErrorCarga generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_LOG\_ERROR\_CARGA"  
)  
public class TxLogErrorCarga implements java.io.Serializable {  
  
  
 private BigDecimal logErrorCargaId;  
 private String tableName;  
 private String errm;  
 private String data;  
 private Date errorDate;  
 private String archivo;  
  
 public TxLogErrorCarga() {  
 }  
  
 public TxLogErrorCarga(String tableName, String errm, String data, Date errorDate, String archivo) {  
 this.tableName = tableName;  
 this.errm = errm;  
 this.data = data;  
 this.errorDate = errorDate;  
 this.archivo = archivo;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="LOG\_ERROR\_CARGA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getLogErrorCargaId() {  
 return this.logErrorCargaId;  
 }  
   
 public void setLogErrorCargaId(BigDecimal logErrorCargaId) {  
 this.logErrorCargaId = logErrorCargaId;  
 }  
  
   
 @Column(name="TABLE\_NAME", nullable=false, length=200)  
 public String getTableName() {  
 return this.tableName;  
 }  
   
 public void setTableName(String tableName) {  
 this.tableName = tableName;  
 }  
  
   
 @Column(name="ERRM", nullable=false, length=4000)  
 public String getErrm() {  
 return this.errm;  
 }  
   
 public void setErrm(String errm) {  
 this.errm = errm;  
 }  
  
   
 @Column(name="DATA", nullable=false, length=4000)  
 public String getData() {  
 return this.data;  
 }  
   
 public void setData(String data) {  
 this.data = data;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="ERROR\_DATE", nullable=false, length=11)  
 public Date getErrorDate() {  
 return this.errorDate;  
 }  
   
 public void setErrorDate(Date errorDate) {  
 this.errorDate = errorDate;  
 }  
  
   
 @Column(name="ARCHIVO", nullable=false, length=2000)  
 public String getArchivo() {  
 return this.archivo;  
 }  
   
 public void setArchivo(String archivo) {  
 this.archivo = archivo;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxPerfilesVariable.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import java.util.Date;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* TxPerfilesVariable generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_PERFILES\_VARIABLE"  
)  
public class TxPerfilesVariable implements java.io.Serializable, IFTxCincominutalable {  
  
  
 private BigDecimal perfilId;  
 private TxCincominutal txCincominutal;  
 private CatVariablePerfil catVariablePerfil;  
 private CatEstatusPerfil catEstatusPerfil;  
 private CatMedidores catMedidores;  
 private BigDecimal principal;  
 private BigDecimal valor;  
 private BigDecimal energiaPc;  
 private Date fechaPc;  
 private BigDecimal energiaSc;  
 private Date fechaSc;  
  
 public TxPerfilesVariable() {  
 }  
  
 public TxPerfilesVariable(TxCincominutal txCincominutal, CatVariablePerfil catVariablePerfil, CatEstatusPerfil catEstatusPerfil, CatMedidores catMedidores, BigDecimal principal, BigDecimal valor) {  
 this.txCincominutal = txCincominutal;  
 this.catVariablePerfil = catVariablePerfil;  
 this.catEstatusPerfil = catEstatusPerfil;  
 this.catMedidores = catMedidores;  
 this.principal = principal;  
 this.valor = valor;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="PERFIL\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getPerfilId() {  
 return this.perfilId;  
 }  
   
 public void setPerfilId(BigDecimal perfilId) {  
 this.perfilId = perfilId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="CINCOMINUTAL\_ID")  
 public TxCincominutal getTxCincominutal() {  
 return this.txCincominutal;  
 }  
   
 public void setTxCincominutal(TxCincominutal txCincominutal) {  
 this.txCincominutal = txCincominutal;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="VARIABLE\_PERFIL\_ID")  
 public CatVariablePerfil getCatVariablePerfil() {  
 return this.catVariablePerfil;  
 }  
   
 public void setCatVariablePerfil(CatVariablePerfil catVariablePerfil) {  
 this.catVariablePerfil = catVariablePerfil;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="ESTATUS\_PERFIL\_ID")  
 public CatEstatusPerfil getCatEstatusPerfil() {  
 return this.catEstatusPerfil;  
 }  
   
 public void setCatEstatusPerfil(CatEstatusPerfil catEstatusPerfil) {  
 this.catEstatusPerfil = catEstatusPerfil;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="MEDIDOR\_ID")  
 public CatMedidores getCatMedidores() {  
 return this.catMedidores;  
 }  
   
 public void setCatMedidores(CatMedidores catMedidores) {  
 this.catMedidores = catMedidores;  
 }  
  
   
 @Column(name="PRINCIPAL", precision=22, scale=0)  
 public BigDecimal getPrincipal() {  
 return this.principal;  
 }  
   
 public void setPrincipal(BigDecimal principal) {  
 this.principal = principal;  
 }  
  
   
 @Column(name="VALOR", precision=22, scale=0)  
 public BigDecimal getValor() {  
 return this.valor;  
 }  
   
 public void setValor(BigDecimal valor) {  
 this.valor = valor;  
 }  
  
  
 @Column(name="ENERGIA\_PC", precision=22, scale=0)  
 public BigDecimal getEnergiaPc() {  
 return this.energiaPc;  
 }  
   
 public void setEnergiaPc(BigDecimal energiaPc) {  
 this.energiaPc = energiaPc;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="FECHA\_PC", length=11)  
 public Date getFechaPc() {  
 return this.fechaPc;  
 }  
   
 public void setFechaPc(Date fechaPc) {  
 this.fechaPc = fechaPc;  
 }  
  
   
 @Column(name="ENERGIA\_SC", precision=22, scale=0)  
 public BigDecimal getEnergiaSc() {  
 return this.energiaSc;  
 }  
   
 public void setEnergiaSc(BigDecimal energiaSc) {  
 this.energiaSc = energiaSc;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name="FECHA\_SC", length=11)  
 public Date getFechaSc() {  
 return this.fechaSc;  
 }  
   
 public void setFechaSc(Date fechaSc) {  
 this.fechaSc = fechaSc;  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxProduccion.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* TxProduccion generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_PRODUCCION"  
)  
public class TxProduccion implements java.io.Serializable {  
  
  
 private BigDecimal produccionId;  
 private CatPropCincominutal catPropCincominutal;  
 private TxCincominutal txCincominutal;  
 private String valor;  
  
 public TxProduccion() {  
 }  
  
 public TxProduccion(CatPropCincominutal catPropCincominutal, TxCincominutal txCincominutal, String valor) {  
 this.catPropCincominutal = catPropCincominutal;  
 this.txCincominutal = txCincominutal;  
 this.valor = valor;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="PRODUCCION\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getProduccionId() {  
 return this.produccionId;  
 }  
   
 public void setProduccionId(BigDecimal produccionId) {  
 this.produccionId = produccionId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="PROPIEDAD\_ID")  
 public CatPropCincominutal getCatPropCincominutal() {  
 return this.catPropCincominutal;  
 }  
   
 public void setCatPropCincominutal(CatPropCincominutal catPropCincominutal) {  
 this.catPropCincominutal = catPropCincominutal;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="CINCOMINUTAL\_ID")  
 public TxCincominutal getTxCincominutal() {  
 return this.txCincominutal;  
 }  
   
 public void setTxCincominutal(TxCincominutal txCincominutal) {  
 this.txCincominutal = txCincominutal;  
 }  
  
   
 @Column(name="VALOR", length=20)  
 public String getValor() {  
 return this.valor;  
 }  
   
 public void setValor(String valor) {  
 this.valor = valor;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxRolPermiso.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* TxRolPermiso generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_ROL\_PERMISO"  
)  
public class TxRolPermiso implements java.io.Serializable {  
  
  
 private BigDecimal rolPermisoId;  
 private CatPermiso catPermiso;  
 private CatRol catRol;  
  
 public TxRolPermiso() {  
 }  
  
   
 public TxRolPermiso(CatRol catRol) {  
 this.catRol = catRol;  
 }  
 public TxRolPermiso(CatPermiso catPermiso, CatRol catRol) {  
 this.catPermiso = catPermiso;  
 this.catRol = catRol;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="ROL\_PERMISO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getRolPermisoId() {  
 return this.rolPermisoId;  
 }  
   
 public void setRolPermisoId(BigDecimal rolPermisoId) {  
 this.rolPermisoId = rolPermisoId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="PERMISO\_ID")  
 public CatPermiso getCatPermiso() {  
 return this.catPermiso;  
 }  
   
 public void setCatPermiso(CatPermiso catPermiso) {  
 this.catPermiso = catPermiso;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="ROL\_ID", nullable=false)  
 public CatRol getCatRol() {  
 return this.catRol;  
 }  
   
 public void setCatRol(CatRol catRol) {  
 this.catRol = catRol;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxTrazabilidad.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import java.util.Date;  
import java.util.HashSet;  
import java.util.Set;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
/\*\*  
 \* TxTrazabilidad generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_TRAZABILIDAD"  
)  
public class TxTrazabilidad implements java.io.Serializable {  
  
  
 private BigDecimal trazabilidadId;  
 private Date fecha;  
 private String accion;  
 private BigDecimal usuarioId;  
 private BigDecimal rolId;  
 private Set<TxTrazaQuery> txTrazaQueries = new HashSet<TxTrazaQuery>(0);  
  
 public TxTrazabilidad() {  
 }  
  
   
 public TxTrazabilidad(Date fecha) {  
 this.fecha = fecha;  
 }  
 public TxTrazabilidad(Date fecha, String accion, BigDecimal usuarioId, BigDecimal rolId, Set<TxTrazaQuery> txTrazaQueries) {  
 this.fecha = fecha;  
 this.accion = accion;  
 this.usuarioId = usuarioId;  
 this.rolId = rolId;  
 this.txTrazaQueries = txTrazaQueries;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="TRAZABILIDAD\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getTrazabilidadId() {  
 return this.trazabilidadId;  
 }  
   
 public void setTrazabilidadId(BigDecimal trazabilidadId) {  
 this.trazabilidadId = trazabilidadId;  
 }  
  
 @Temporal(TemporalType.DATE)  
 @Column(name="FECHA", nullable=false, length=7)  
 public Date getFecha() {  
 return this.fecha;  
 }  
   
 public void setFecha(Date fecha) {  
 this.fecha = fecha;  
 }  
  
   
 @Column(name="ACCION", length=100)  
 public String getAccion() {  
 return this.accion;  
 }  
   
 public void setAccion(String accion) {  
 this.accion = accion;  
 }  
  
   
 @Column(name="USUARIO\_ID", precision=22, scale=0)  
 public BigDecimal getUsuarioId() {  
 return this.usuarioId;  
 }  
   
 public void setUsuarioId(BigDecimal usuarioId) {  
 this.usuarioId = usuarioId;  
 }  
  
   
 @Column(name="ROL\_ID", precision=22, scale=0)  
 public BigDecimal getRolId() {  
 return this.rolId;  
 }  
   
 public void setRolId(BigDecimal rolId) {  
 this.rolId = rolId;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="txTrazabilidad")  
 public Set<TxTrazaQuery> getTxTrazaQueries() {  
 return this.txTrazaQueries;  
 }  
   
 public void setTxTrazaQueries(Set<TxTrazaQuery> txTrazaQueries) {  
 this.txTrazaQueries = txTrazaQueries;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxTrazaQuery.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* TxTrazaQuery generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_TRAZA\_QUERY"  
)  
public class TxTrazaQuery implements java.io.Serializable {  
  
  
 private BigDecimal trazaQueryId;  
 private CatQuery catQuery;  
 private TxTrazabilidad txTrazabilidad;  
  
 public TxTrazaQuery() {  
 }  
  
 public TxTrazaQuery(CatQuery catQuery, TxTrazabilidad txTrazabilidad) {  
 this.catQuery = catQuery;  
 this.txTrazabilidad = txTrazabilidad;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="TRAZA\_QUERY\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getTrazaQueryId() {  
 return this.trazaQueryId;  
 }  
   
 public void setTrazaQueryId(BigDecimal trazaQueryId) {  
 this.trazaQueryId = trazaQueryId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="QUERY\_ID", nullable=false)  
 public CatQuery getCatQuery() {  
 return this.catQuery;  
 }  
   
 public void setCatQuery(CatQuery catQuery) {  
 this.catQuery = catQuery;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="TRAZABILIDAD\_ID", nullable=false)  
 public TxTrazabilidad getTxTrazabilidad() {  
 return this.txTrazabilidad;  
 }  
   
 public void setTxTrazabilidad(TxTrazabilidad txTrazabilidad) {  
 this.txTrazabilidad = txTrazabilidad;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxUsuario.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import java.util.HashSet;  
import java.util.Set;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.persistence.UniqueConstraint;  
  
/\*\*  
 \* TxUsuario generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_USUARIO"  
 , uniqueConstraints = @UniqueConstraint(columnNames="USERNAME")   
)  
public class TxUsuario implements java.io.Serializable {  
  
  
 private BigDecimal usuarioId;  
 private String username;  
 private String password;  
 private Set<TxUsuarioPlanta> txUsuarioPlantas = new HashSet<TxUsuarioPlanta>(0);  
 private Set<TxUsuarioRol> txUsuarioRols = new HashSet<TxUsuarioRol>(0);  
 private Set<TxDocumentos> txDocumentoses = new HashSet<TxDocumentos>(0);  
  
 public TxUsuario() {  
 }  
  
   
 public TxUsuario(String username, String password) {  
 this.username = username;  
 this.password = password;  
 }  
 public TxUsuario(String username, String password, Set<TxUsuarioPlanta> txUsuarioPlantas, Set<TxUsuarioRol> txUsuarioRols, Set<TxDocumentos> txDocumentoses) {  
 this.username = username;  
 this.password = password;  
 this.txUsuarioPlantas = txUsuarioPlantas;  
 this.txUsuarioRols = txUsuarioRols;  
 this.txDocumentoses = txDocumentoses;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="USUARIO\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getUsuarioId() {  
 return this.usuarioId;  
 }  
   
 public void setUsuarioId(BigDecimal usuarioId) {  
 this.usuarioId = usuarioId;  
 }  
  
   
 @Column(name="USERNAME", unique=true, nullable=false, length=150)  
 public String getUsername() {  
 return this.username;  
 }  
   
 public void setUsername(String username) {  
 this.username = username;  
 }  
  
   
 @Column(name="PASSWORD", nullable=false, length=200)  
 public String getPassword() {  
 return this.password;  
 }  
   
 public void setPassword(String password) {  
 this.password = password;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="txUsuario")  
 public Set<TxUsuarioPlanta> getTxUsuarioPlantas() {  
 return this.txUsuarioPlantas;  
 }  
   
 public void setTxUsuarioPlantas(Set<TxUsuarioPlanta> txUsuarioPlantas) {  
 this.txUsuarioPlantas = txUsuarioPlantas;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="txUsuario")  
 public Set<TxUsuarioRol> getTxUsuarioRols() {  
 return this.txUsuarioRols;  
 }  
   
 public void setTxUsuarioRols(Set<TxUsuarioRol> txUsuarioRols) {  
 this.txUsuarioRols = txUsuarioRols;  
 }  
  
@OneToMany(fetch=FetchType.LAZY, mappedBy="txUsuario")  
 public Set<TxDocumentos> getTxDocumentoses() {  
 return this.txDocumentoses;  
 }  
   
 public void setTxDocumentoses(Set<TxDocumentos> txDocumentoses) {  
 this.txDocumentoses = txDocumentoses;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxUsuarioPlanta.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* TxUsuarioPlanta generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_USUARIO\_PLANTA"  
)  
public class TxUsuarioPlanta implements java.io.Serializable {  
  
  
 private BigDecimal usuarioPlantaId;  
 private CatPlanta catPlanta;  
 private TxUsuario txUsuario;  
  
 public TxUsuarioPlanta() {  
 }  
  
 public TxUsuarioPlanta(CatPlanta catPlanta, TxUsuario txUsuario) {  
 this.catPlanta = catPlanta;  
 this.txUsuario = txUsuario;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="USUARIO\_PLANTA\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getUsuarioPlantaId() {  
 return this.usuarioPlantaId;  
 }  
   
 public void setUsuarioPlantaId(BigDecimal usuarioPlantaId) {  
 this.usuarioPlantaId = usuarioPlantaId;  
 }  
  
@ManyToOne(fetch=FetchType.EAGER)  
 @JoinColumn(name="PLANTA\_ID")  
 public CatPlanta getCatPlanta() {  
 return this.catPlanta;  
 }  
   
 public void setCatPlanta(CatPlanta catPlanta) {  
 this.catPlanta = catPlanta;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="USUARIO\_ID")  
 public TxUsuario getTxUsuario() {  
 return this.txUsuario;  
 }  
   
 public void setTxUsuario(TxUsuario txUsuario) {  
 this.txUsuario = txUsuario;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxUsuarioRol.java

package mx.mbi.dgc.jpa.entity;  
// Generated 12/10/2018 01:26:41 PM by Hibernate Tools 5.2.11.Final  
  
  
import java.math.BigDecimal;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
  
/\*\*  
 \* TxUsuarioRol generated by hbm2java  
 \*/  
@Entity  
@Table(name="TX\_USUARIO\_ROL"  
)  
public class TxUsuarioRol implements java.io.Serializable {  
  
  
 private BigDecimal usuarioRolId;  
 private CatRol catRol;  
 private TxUsuario txUsuario;  
  
 public TxUsuarioRol() {  
 }  
  
 public TxUsuarioRol(CatRol catRol, TxUsuario txUsuario) {  
 this.catRol = catRol;  
 this.txUsuario = txUsuario;  
 }  
   
 @Id @GeneratedValue(strategy=IDENTITY)  
  
   
 @Column(name="USUARIO\_ROL\_ID", unique=true, nullable=false, precision=22, scale=0)  
 public BigDecimal getUsuarioRolId() {  
 return this.usuarioRolId;  
 }  
   
 public void setUsuarioRolId(BigDecimal usuarioRolId) {  
 this.usuarioRolId = usuarioRolId;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="ROL\_ID", nullable=false)  
 public CatRol getCatRol() {  
 return this.catRol;  
 }  
   
 public void setCatRol(CatRol catRol) {  
 this.catRol = catRol;  
 }  
  
@ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name="USUARIO\_ID", nullable=false)  
 public TxUsuario getTxUsuario() {  
 return this.txUsuario;  
 }  
   
 public void setTxUsuario(TxUsuario txUsuario) {  
 this.txUsuario = txUsuario;  
 }  
  
  
  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\jpa\entity\TxValorFActuracion.java

package mx.mbi.dgc.jpa.entity;  
  
import java.math.BigDecimal;  
  
import java.util.Date;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.FetchType;  
import javax.persistence.GeneratedValue;  
import static javax.persistence.GenerationType.IDENTITY;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;  
import javax.persistence.Temporal;  
import javax.persistence.TemporalType;  
  
@Entity  
@Table(name="TX\_VALOR\_FACTURACION")  
public class TxValorFActuracion {  
  
 private Integer valorFacturacionId;  
 private CatVariableFacturacion variableFacturacion;  
 private BigDecimal valor;  
 private Date fecha;  
  
 public TxValorFActuracion(){  
 }  
  
 @Id  
 @GeneratedValue(strategy=IDENTITY)  
  
 @Column(name="VALOR\_FACTURACION\_ID", nullable = false, unique = true)  
 public Integer getValorFacturacionId() {  
 return valorFacturacionId;  
 }  
  
 public void setValorFacturacionId(Integer valorFacturacionId) {  
 this.valorFacturacionId = valorFacturacionId;  
 }  
  
 @ManyToOne(fetch=FetchType.LAZY)  
 @JoinColumn(name = "VARIABLE\_FACTURACION\_ID", nullable = false)  
 public CatVariableFacturacion getVariableFacturacion() {  
 return variableFacturacion;  
 }  
  
 public void setVariableFacturacion(CatVariableFacturacion variableFacturacion) {  
 this.variableFacturacion = variableFacturacion;  
 }  
  
 @Column(name = "VALOR")  
 public BigDecimal getValor() {  
 return valor;  
 }  
  
 public void setValor(BigDecimal valor) {  
 this.valor = valor;  
 }  
  
 @Temporal(TemporalType.TIMESTAMP)  
 @Column(name = "FECHA")  
 public Date getFecha() {  
 return fecha;  
 }  
  
 public void setFecha(Date fecha) {  
 this.fecha = fecha;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\ClienteConfig.java

package mx.mbi.dgc.microservices;  
  
import java.util.concurrent.Executor;  
  
import mx.mbi.dgc.excepcion.CustomAsyncExceptionHandler;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.aop.interceptor.AsyncUncaughtExceptionHandler;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.core.task.AsyncTaskExecutor;  
import org.springframework.core.task.SimpleAsyncTaskExecutor;  
import org.springframework.oxm.jaxb.Jaxb2Marshaller;  
import org.springframework.scheduling.annotation.AsyncConfigurer;  
import org.springframework.scheduling.annotation.EnableAsync;  
import org.springframework.scheduling.annotation.EnableScheduling;  
import org.springframework.scheduling.concurrent.ThreadPoolTaskExecutor;  
import org.springframework.scheduling.concurrent.ThreadPoolTaskScheduler;  
import org.springframework.ws.client.core.WebServiceTemplate;  
  
@EnableScheduling  
@EnableAsync  
@Configuration  
public class ClienteConfig implements AsyncConfigurer {  
 private static final Logger LOG = LogManager.getLogger(ClienteConfig.class);  
  
 @Bean(name = "threadPoolTaskExecutor")  
 public Executor threadPoolTaskExecutor() {  
 return new ThreadPoolTaskExecutor();  
 }  
   
 @Override  
 public Executor getAsyncExecutor() {  
 ThreadPoolTaskScheduler scheduler = new ThreadPoolTaskScheduler();  
 scheduler.setPoolSize(20);  
 scheduler.initialize();  
 return scheduler;  
 }  
   
 @Override  
 public AsyncUncaughtExceptionHandler getAsyncUncaughtExceptionHandler() {  
 return new CustomAsyncExceptionHandler();  
 }  
   
 @Bean(name = "simpleAsyncTaskExecutor")  
 public AsyncTaskExecutor taskExecutor () {  
 SimpleAsyncTaskExecutor t = new SimpleAsyncTaskExecutor();  
 t.setConcurrencyLimit(20);  
 return t;  
 }  
   
  
 @Bean  
 Jaxb2Marshaller jaxb2Marshaller() {  
 Jaxb2Marshaller jaxb2Marshaller = new Jaxb2Marshaller();  
 jaxb2Marshaller.setPackagesToScan("mx.mbi.dgc.dominio");  
 return jaxb2Marshaller;  
 }  
  
 @Bean  
 public WebServiceTemplate webServiceTemplate() {  
 LOG.info("execute ClienteConfig.webServiceTemplate. . .");  
 WebServiceTemplate webServiceTemplate = new WebServiceTemplate();  
 webServiceTemplate.setMarshaller(jaxb2Marshaller());  
 webServiceTemplate.setUnmarshaller(jaxb2Marshaller());  
 return webServiceTemplate;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\negocio\ControllerCapturaValoresCargos.java

package mx.mbi.dgc.microservices.controller.negocio;  
  
import mx.mbi.dgc.business.negocio.IfaceCapturaValoresCargos;  
import mx.mbi.dgc.dominio.negocio.DominioCapturaValoresCargos;  
import mx.mbi.dgc.dominio.negocio.DominioDetalleCVC;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CapturaValoresCargos")  
public class ControllerCapturaValoresCargos {  
 private static final Logger LOG = LogManager.getLogger(ControllerCapturaValoresCargos.class);  
 @Autowired  
 private IfaceCapturaValoresCargos businessCapturaValoresCargos;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/getCargos",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCapturaValoresCargos getCargos(@RequestBody DominioCapturaValoresCargos dom) throws Exception {  
 dom.setStatus(false);  
 dom.setCode(500);  
 try {  
 LOG.info("invoke ControllerCapturaValoresCargos.getCargos. . .");  
 dom = businessCapturaValoresCargos.getCargos(dom.getCatPlanta().getPlantaId());  
 } catch(Exception ex) {  
 LOG.error("Error in invoke ControllerCapturaValoresCargos.getCargos: "+ex);  
 dom.setMsg("Error in invoke ControllerCapturaValoresCargos.getCargos: "+ex);  
 }  
 return dom;  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/cargaEvidencia",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCapturaValoresCargos cargaEvidencia(@RequestBody DominioCapturaValoresCargos dom) throws Exception {  
 dom.setStatus(false);  
 dom.setCode(500);  
 try {  
 LOG.info("invoke ControllerCapturaValoresCargos.cargaEvidencia. . .");  
 dom = businessCapturaValoresCargos.cargaEvidencia(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke ControllerCapturaValoresCargos.cargaEvidencia: "+ex);  
 dom.setMsg("Error in invoke ControllerCapturaValoresCargos.cargaEvidencia: "+ex);  
 }  
 return dom;  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/getDetalle",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCapturaValoresCargos getDetalle(@RequestBody DominioCapturaValoresCargos dom) throws Exception {  
 dom.setStatus(false);  
 dom.setCode(500);  
 try {  
 LOG.info("invoke ControllerCapturaValoresCargos.getDetalle. . .");  
 dom = businessCapturaValoresCargos.getDetalle(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke ControllerCapturaValoresCargos.getDetalle: "+ex);  
 dom.setMsg("Error in invoke ControllerCapturaValoresCargos.getDetalle: "+ex);  
 }  
 return dom;  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/downloadFile",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioDetalleCVC downloadFile(@RequestBody DominioDetalleCVC dom) throws Exception {  
 try {  
 LOG.info("invoke ControllerCapturaValoresCargos.downloadFile. . .");  
 dom = businessCapturaValoresCargos.getFilePath(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke ControllerCapturaValoresCargos.downloadFile: "+ex);  
 }  
 return dom;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\negocio\ControllerCargaInformacion.java

package mx.mbi.dgc.microservices.controller.negocio;  
  
import mx.mbi.dgc.business.carga.IfaceCargaInformacion;  
import mx.mbi.dgc.dominio.DominioCargaInformacion;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CargaInformacion")  
public class ControllerCargaInformacion {  
 private static final Logger LOG = LogManager.getLogger(ControllerCargaInformacion.class);  
 @Autowired  
 private IfaceCargaInformacion businessCargaInformacion;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/cargaCSV",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCargaInformacion cargaCSV(@RequestBody DominioCargaInformacion dominio) throws Exception {  
 dominio.setStatus(false);  
 dominio.setCode(500);  
 try {  
 System.out.println("invoke ControllerCargaInformacion.cargaCSV: "+dominio.getFileName()+" - User: "+dominio.getIdUsuario());  
 System.out.println("invoke ControllerCargaInformacion.cargaCSV: "+dominio.getFileName()+" - planat\_id: "+dominio.getIdPlanta());  
 boolean status = businessCargaInformacion.cargaCSV(dominio.getFileName(), dominio.getIdUsuario(),dominio.getIdPlanta());  
 if(status) {  
 LOG.info("Carga success from file: "+dominio.getFileName());  
 dominio.setMsg("success");  
 dominio.setStatus(true);  
 dominio.setCode(200);  
 } else {  
 LOG.error("Error in invoke boolean.BusinessCargaInformacion.cargaCSV.call: "+dominio.getFileName());  
 }  
 } catch(Exception ex) {  
 LOG.error("Error in invoke ControllerCargaInformacion.cargaCSV: "+ex);  
 dominio.setMsg("Error in invoke ControllerCargaInformacion.cargaCSV: "+ex);  
 }  
 return dominio;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\negocio\ControllerCargaInformacionPerfiles.java

package mx.mbi.dgc.microservices.controller.negocio;  
  
import mx.mbi.dgc.business.carga.IfaceCargaInformacionPerfiles;  
import mx.mbi.dgc.dominio.DominioCargaInformacion;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CargaInformacionPerfiles")  
public class ControllerCargaInformacionPerfiles {  
 private static final Logger LOG = LogManager.getLogger(ControllerCargaInformacionPerfiles.class);  
   
 @Autowired  
 private IfaceCargaInformacionPerfiles businessCargaInformacionPerfiles;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/cargaCSV",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCargaInformacion cargaCSV(@RequestBody DominioCargaInformacion dominio) throws Exception {  
 dominio.setStatus(false);  
 dominio.setCode(500);  
 try {  
 System.out.println("invoke ControllerCargaInformacionPerfiles.cargaCSV: "+dominio.getFileName()+" - User: "+dominio.getIdUsuario());  
 dominio = businessCargaInformacionPerfiles.cargaCSV(dominio);  
 if(dominio.getCode() == 200) {  
 System.out.println("Carga success from file: "+dominio.getFileName());  
 dominio.setMsg("success");  
 dominio.setStatus(true);  
 dominio.setCode(200);  
 } else {  
 System.out.println("Error in invoke boolean.BusinessCargaInformacion.cargaCSV.call: "+dominio.getFileName());  
 }  
 } catch(Exception ex) {  
 ex.printStackTrace();  
 dominio.setMsg("Error in invoke ControllerCargaInformacion.cargaCSV: "+ex);  
 }  
 return dominio;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\negocio\ControllerModeloMatematico.java

package mx.mbi.dgc.microservices.controller.negocio;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.mm.IfaceModeloMatematico;  
import mx.mbi.dgc.dominio.DominioTxCincominutalVar;  
import mx.mbi.dgc.dominio.negocio.DominioModeloMatematico;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.jpa.entity.TxCincominutalVar;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/ModeloMatematico")  
public class ControllerModeloMatematico {  
 private static final Logger LOG = LogManager.getLogger(ControllerModeloMatematico.class);  
 @Autowired  
 private IfaceModeloMatematico businessModeloMatematico;  
  
 @RequestMapping(method = RequestMethod.POST, path = "/calculaFactores", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioModeloMatematico calculaFactores(@RequestBody DominioModeloMatematico dom) throws DyCNormaException {  
 dom.setStatus(false);  
 dom.setCode(500);  
 try {  
 System.out.println("invoke ControllerModeloMatematico.calculaFactores dom.getTipoCalculo(): " +  
 dom.getTipoCalculo());  
 return businessModeloMatematico.calculaFactores(dom);  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 LOG.error("Error in invoke ControllerModeloMatematico.calculaFactores: " + ex);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/actualizaMM", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioModeloMatematico actualizaMM(@RequestBody DominioModeloMatematico dom) throws DyCNormaException {  
 dom.setStatus(false);  
 dom.setCode(500);  
 try {  
 System.out.println("invoke ControllerModeloMatematico.actualizaMM: " + dom.getTipoCalculo());  
 List<TxCincominutalVar> lista = new ArrayList<>();  
 for (DominioTxCincominutalVar dom2 : dom.getListaCincominutal()) {  
 TxCincominutalVar entity = new TxCincominutalVar();  
 JRUtil.copyProperties(entity, dom2);  
 lista.add(entity);  
 }  
 businessModeloMatematico.actualizaMM(dom.getDominioDyCNorma(), lista);  
  
 dom.setCode(200);  
 return dom;  
  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 LOG.error("Error in invoke ControllerModeloMatematico.calculaFactores: " + ex);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\negocio\ControllerMonitorFacturaCargosFijos.java

package mx.mbi.dgc.microservices.controller.negocio;  
  
import mx.mbi.dgc.business.facturacion.IfaceMontosFactura;  
import mx.mbi.dgc.dominio.negocio.facturacion.calculos.DominioCalculosCargosFijos;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CalculosDashBoardFactura")  
public class ControllerMonitorFacturaCargosFijos {  
 private static final Logger LOG = LogManager.getLogger(ControllerMonitorFacturaCargosFijos.class);  
 @Autowired  
 private IfaceMontosFactura businessMontosFactura;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/getCalculosCargoFijo",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCalculosCargosFijos getCalculosPCFC(@RequestBody DominioCalculosCargosFijos dom) throws Exception {  
 dom.setStatus(true);  
 dom.setCode(200);  
 try {  
 System.out.println("invoke ControllerCapturaValoresCargos.getCargos. . .");  
 //dom = businessMontosFactura.montosPCFOM(dom);  
 dom = businessMontosFactura.montosPCFC(dom);  
 } catch(Exception ex) {  
 ex.printStackTrace();  
 dom.setStatus(false);  
 dom.setCode(500);  
 System.out.println("Error in invoke ControllerCapturaValoresCargos.getCargos: "+ex);  
 dom.setMsg("Error in invoke ControllerCapturaValoresCargos.getCargos: "+ex);  
 }  
 return dom;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\negocio\facturacion\RESTFacturacion.java

package mx.mbi.dgc.microservices.controller.negocio.facturacion;  
  
import java.math.BigDecimal;  
  
import java.text.SimpleDateFormat;  
  
import java.time.YearMonth;  
  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceTxDocumentos;  
import mx.mbi.dgc.business.facturacion.IfaceBusinessFacturacion;  
import mx.mbi.dgc.business.facturacion.facturador.Facturador;  
import mx.mbi.dgc.dominio.DominioCatCompaniaPlantas;  
import mx.mbi.dgc.dominio.DominioCatPlanta;  
import mx.mbi.dgc.dominio.DominioCatPlantaTipoFacturas;  
import mx.mbi.dgc.dominio.DominioCatServicioImpuestos;  
import mx.mbi.dgc.dominio.DominioCatServicios;  
import mx.mbi.dgc.dominio.DominioTxDocumentos;  
import mx.mbi.dgc.dominio.negocio.DominioFacturas;  
import mx.mbi.dgc.dominio.negocio.facturacion.DominioFacturacion;  
import mx.mbi.dgc.global.Globales;  
import mx.mbi.dgc.microservices.controller.RESTCatEstatusPerfil;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
import org.springframework.web.client.RestTemplate;  
  
@RestController  
@RequestMapping(path = "/servicesREST/facturacion")  
public class RESTFacturacion {  
 private static final Logger LOG = LogManager.getLogger(RESTCatEstatusPerfil.class);  
 @Autowired  
 private IfaceBusinessFacturacion businessFacturacion;  
 @Autowired  
 private IfaceTxDocumentos businessTxDocumentos;  
  
  
 @RequestMapping(method = RequestMethod.POST, path = "/getMaxFolioBySerie", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 String findById(@RequestBody DominioTxDocumentos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTFacturacion.getMaxFolioBySerie. . .");  
  
 return String.valueOf(businessFacturacion.getMaxFolioBySerie(dom));  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTFacturacion.getMaxFolioBySerie: " + ex);  
  
 return "-1";  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findCompaniaPlantaByPlantaId", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatCompaniaPlantas findCompaniaPlantaByPlantaId(@RequestBody DominioFacturacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlantaTipoFacturas.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 DominioCatCompaniaPlantas ret = businessFacturacion.findCompaniaPlantaByPlantaId(dom);  
 return ret;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatPlantaTipoFacturas.findById: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return new DominioCatCompaniaPlantas();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findImpuestosByServicio", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioCatServicioImpuestos> findByCatPlanta(@RequestBody DominioCatServicios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTFacturacion.findImpuestosByServicio. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 List<DominioCatServicioImpuestos> lista = businessFacturacion.findImpuestosByServicio(dom);  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTFacturacion.findImpuestosByServicio... " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findTipoFacturasByCatPlanta", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioCatPlantaTipoFacturas> findByCatPlanta(@RequestBody DominioCatPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlantaTipoFacturas.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 List<DominioCatPlantaTipoFacturas> lista = businessFacturacion.findTipoFacturasByCatPlanta(dom);  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatPlantaTipoFacturas.findById: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/sellarYTimbrarFactura", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxDocumentos sellarYTimbrarFactura(@RequestBody DominioTxDocumentos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlantaTipoFacturas.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 RestTemplate restTemplate = new RestTemplate();  
 DominioTxDocumentos result =  
 restTemplate.postForObject("https://<%=request.getServerName()%>:<%=request.getServerPort()%>safe/dgc/servicesREST/facturacion/cadenaOriginal",  
 dom, DominioTxDocumentos.class);  
 dom = (DominioTxDocumentos) businessTxDocumentos.update(result);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke sellarYTimbrarFactura.findById: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return new DominioTxDocumentos();  
 }  
 }  
  
  
 @RequestMapping(method = RequestMethod.POST, path = "/crearFactura", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioFacturas crearFactura(@RequestBody DominioFacturas peticion) {  
 try {  
 SimpleDateFormat formatoFecha = new SimpleDateFormat("dd/MM/yyyy");  
  
 DominioFacturas respuesta = new DominioFacturas();  
 peticion.setStatus(false);  
 peticion.setCode(500);  
  
 System.out.printf("Mes a facturar: %d\nAnio: %d\n", peticion.getMes(), peticion.getAnio());  
 YearMonth mes = YearMonth.of(peticion.getAnio(), peticion.getMes());  
 Globales.Plantas planta = Globales.getPlanta(peticion.getPlanta());  
 Date fechaDePago = formatoFecha.parse(peticion.getFechaDePago());  
 Facturador.TipoDeFactura tipoDeFactura = Facturador.getTipoDeFactura(peticion.getTipoDeFactura());  
 HashMap<String, BigDecimal> varsMap = new HashMap<>();  
 for (String var : peticion.getVariablesManuales()) {  
 String[] keyVar = var.split(":");  
 varsMap.put(keyVar[0], new BigDecimal(keyVar[1]));  
 }  
 Facturador facturador = new Facturador(tipoDeFactura, varsMap);  
 respuesta.setPathDescarga(facturador.facturar(planta, mes, fechaDePago));  
 respuesta.setTipoDeFactura(tipoDeFactura.getNombre());  
 respuesta.setMonto(facturador.getMonto().toString());  
 System.out.println("FACTURA CREADA!");  
 return respuesta;  
 } catch (Exception e) {  
 e.printStackTrace();  
 DominioFacturas respuesta = new DominioFacturas();  
 respuesta.setCode(500);  
 respuesta.setMsg("Datos insuficientes para crear la factura");  
 return respuesta;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\negocio\facturacion\RESTFacturacionTimbradoSellado.java

package mx.mbi.dgc.microservices.controller.negocio.facturacion;  
  
import java.io.IOException;  
import java.io.StringReader;  
  
import java.text.SimpleDateFormat;  
  
import java.util.Date;  
  
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;  
import javax.xml.parsers.ParserConfigurationException;  
  
import mx.mbi.dgc.business.facturacion.IfaceFacturacionSelladoTimbrado;  
import mx.mbi.dgc.dominio.DominioTxDocumentos;  
import mx.mbi.dgc.excepcion.DyCNormaException;  
import mx.mbi.dgc.global.Globales;  
import mx.mbi.dgc.jpa.entity.TxDocumentos;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.w3c.dom.Document;  
import org.w3c.dom.Element;  
  
import org.xml.sax.InputSource;  
import org.xml.sax.SAXException;  
  
@RestController  
@RequestMapping(path = "/servicesREST/facturacion")  
public class RESTFacturacionTimbradoSellado {  
 private static final Logger LOG = LogManager.getLogger(RESTFacturacionTimbradoSellado.class);  
 @Autowired  
 private IfaceFacturacionSelladoTimbrado facturacionSelladoTimbrado;  
  
  
 @RequestMapping(method = RequestMethod.POST, path = "/cadenaOriginal", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxDocumentos generaCadenaOriginal(@RequestBody DominioTxDocumentos dom) throws Exception {  
 // DominioTxDocumentos response = new DominioTxDocumentos();  
 dom.setStatus(true);  
 dom.setCode(200);  
 try {  
 LOG.info("invoke RESTFacturacionTimbradoSellado.generaCadenaOriginal. . .");  
  
 //TxDocumentos txDocumentos = new TxDocumentos();  
 //JRUtil.copyProperties(txDocumentos, dom);  
 dom.setCadenaOriginal(facturacionSelladoTimbrado.cfdiCadenaOriginal(dom));  
 //JRUtil.copyProperties(dom, txDocumentos);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTFacturacionTimbradoSellado.generaCadenaOrigina: " + ex);  
 ex.printStackTrace();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCadenaOriginal(null);  
 return dom;  
 }  
 return dom;  
 }  
  
  
 // public String creaCertificado(String file\_certificado);  
 @RequestMapping(method = RequestMethod.POST, path = "/certificado", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxDocumentos generaCertificado(@RequestBody DominioTxDocumentos dom) throws Exception {  
  
 dom.setStatus(true);  
 dom.setCode(200);  
 try {  
 TxDocumentos txDocumentos = new TxDocumentos();  
 JRUtil.copyProperties(txDocumentos, dom);  
 String rutaCertificado =  
 System.getProperty("os.name")  
 .toUpperCase()  
 .contains("LINUX") ? Globales.RUTA\_CERTIFICADO\_LINUX :  
 Globales.RUTA\_CERTIFICADO\_WINDOWS;  
 rutaCertificado += dom.getCatCompaniaPlantas()  
 .getCatDatosCertificados()  
 .getArchivoCertificado();  
 txDocumentos.setCertificado(facturacionSelladoTimbrado.creaCertificado(rutaCertificado));  
 JRUtil.copyProperties(dom, txDocumentos);  
 } catch (Exception ex) {  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setMsg("Error en la generacion del certificado");  
 dom.setCertificado(null);  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en RESTFacturacionTimbradoSellado.generaCertificado: " + ex);  
 }  
 return dom;  
 }  
  
 //public String creaSello(String cadenaOriginal, String ruta\_llave\_privada\_key, String password);  
 @RequestMapping(method = RequestMethod.POST, path = "/selloDigital", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxDocumentos generaSelloDigital(@RequestBody DominioTxDocumentos dom) throws Exception {  
  
 dom.setStatus(true);  
 dom.setCode(200);  
 try {  
 // TxDocumentos txDocumentos = new TxDocumentos();  
 // JRUtil.copyProperties(txDocumentos, dom);  
 dom.setSelloDigitalEmisor(facturacionSelladoTimbrado.creaSello(dom.getCadenaOriginal(), dom.getCatCompaniaPlantas()  
 .getCatDatosCertificados()  
 .getArchivoKey(),  
 dom.getCatCompaniaPlantas()  
 .getCatDatosCertificados()  
 .getContrasenaKey()));  
 // JRUtil.copyProperties(dom, txDocumentos);  
  
 } catch (Exception ex) {  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setMsg("Error al generar el sello");  
 dom.setCertificado(null);  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en RESTFacturacionTimbradoSellado.generaSelloDigital: " + ex);  
 }  
 return dom;  
 }  
  
 // public Document cfdiXML(TxDocumentos cfdi);  
 @RequestMapping(method = RequestMethod.POST, path = "/xmlDigital", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxDocumentos generaXmlDigital(@RequestBody DominioTxDocumentos dom) throws Exception {  
  
 dom.setStatus(true);  
 dom.setCode(200);  
 try {  
 // TxDocumentos txDocumentos = new TxDocumentos();  
  
 // JRUtil.copyProperties(txDocumentos, dom);  
 dom.setStrXml(facturacionSelladoTimbrado.cfdiXML(dom));  
 // JRUtil.copyProperties(dom, txDocumentos);  
 } catch (Exception ex) {  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setMsg("Error al generar el XML");  
 dom.setStrXml(null);  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en RESTFacturacionTimbradoSellado.generaXmlDigital: " + ex);  
 }  
 return dom;  
 }  
  
 // public String timbra(byte[] str\_xml);  
 @RequestMapping(method = RequestMethod.POST, path = "/timbradoDigital", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxDocumentos generaTimbradoDigital(@RequestBody DominioTxDocumentos dom) throws Exception {  
  
 dom.setStatus(true);  
 dom.setCode(200);  
 try {  
 TxDocumentos txDocumentos = new TxDocumentos();  
 JRUtil.copyProperties(txDocumentos, dom);  
 txDocumentos.setTimbrado(facturacionSelladoTimbrado.timbra(txDocumentos.getStrXml().getBytes()));  
 txDocumentos = obtenerDatosTimbrado(txDocumentos);  
 JRUtil.copyProperties(dom, txDocumentos);  
 } catch (Exception ex) {  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setMsg("Error al generar el timbrado");  
 dom.setStrXml(null);  
 ex.printStackTrace();  
 throw new DyCNormaException("Error en RESTFacturacionTimbradoSellado.generaTimbradoDigital: " + ex);  
 }  
 return dom;  
 }  
  
 private TxDocumentos obtenerDatosTimbrado(TxDocumentos txDocumentos) throws SAXException, IOException,  
 ParserConfigurationException {  
 try {  
 DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();  
 DocumentBuilder builder;  
  
 builder = factory.newDocumentBuilder();  
 Document document = builder.parse(new InputSource(new StringReader(txDocumentos.getTimbrado())));  
 Element rootElement = document.getDocumentElement();  
 txDocumentos.setUuidSat(rootElement.getAttribute("UUID"));  
 System.out.println("Fecha timbrado : " + rootElement.getAttribute("FechaTimbrado"));  
  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss");  
  
 Date d = sdf.parse(rootElement.getAttribute("FechaTimbrado"));  
  
 txDocumentos.setFechaHoraSellado(d);  
  
 txDocumentos.setConfirmacionSat(rootElement.getAttribute("Leyenda"));  
 txDocumentos.setSelloDigitaSat(rootElement.getAttribute("SelloSAT"));  
 txDocumentos.setNoCertificadoSat(rootElement.getAttribute("NoCertificadoSAT"));  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return txDocumentos;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\negocio\RESTDyCNorma.java

package mx.mbi.dgc.microservices.controller.negocio;  
  
import java.math.BigDecimal;  
  
import java.text.DateFormat;  
import java.text.SimpleDateFormat;  
  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
import java.util.StringTokenizer;  
  
import mx.mbi.dgc.business.IfaceCatTags;  
import mx.mbi.dgc.business.dyc.IfaceCorreccionNorma;  
import mx.mbi.dgc.business.dyc.IfaceDeteccionNorma;  
import mx.mbi.dgc.business.dyc.IfaceDyCNorma;  
import mx.mbi.dgc.dao.negocio.tags.TagsInterface;  
import mx.mbi.dgc.dominio.DominioCatEstatusVariable;  
import mx.mbi.dgc.dominio.DominioCatTags;  
import mx.mbi.dgc.dominio.DominioCatVariable;  
import mx.mbi.dgc.dominio.DominioDyCNorma;  
import mx.mbi.dgc.dominio.DominioReporteDyC;  
import mx.mbi.dgc.dominio.DominioTags;  
import mx.mbi.dgc.dominio.negocio.dyc.HashTags;  
import mx.mbi.dgc.dominio.negocio.dyc.TagKey;  
import mx.mbi.dgc.dominio.negocio.dyc.TagValue;  
import mx.mbi.dgc.jpa.entity.CatTags;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
/\*\*  
 \* recibe la informacion de la capa web y llama a los metodos necesarios de la capa de negocios  
 \*/  
  
@RestController  
@RequestMapping(path = "/servicesREST/DyCNorma")  
public class RESTDyCNorma {  
 private static final Logger LOG = LogManager.getLogger(RESTDyCNorma.class);  
 @Autowired  
 private IfaceDeteccionNorma businessIfaceDeteccionNorma; //= new BusinessDeteccionNorma();  
 @Autowired  
 private IfaceCorreccionNorma businessIfaceCorreccionNorma;  
 @Autowired  
 private IfaceDyCNorma businessIfaceDyCNorma;  
 @Autowired  
 private TagsInterface tagsInterface;  
 @Autowired  
 private IfaceCatTags businessCatTags;  
  
 @RequestMapping(method = RequestMethod.POST, path = "/detectaErrores", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioDyCNorma detectaErrores(@RequestBody DominioDyCNorma dom) throws Exception {  
 dom.setStatus(false);  
 dom.setCode(500);  
 try {  
 System.out.println("invoke RESTDyCNorma.detectaErrores. . .");  
 dom = businessIfaceDeteccionNorma.detectaErrores(dom);  
 } catch (Exception ex) {  
 System.out.println(ex.getMessage());  
 ex.printStackTrace();  
 System.out.println("Error in invoke RESTDyCNorma.detectaErrores: " + ex);  
 dom.setMsg("Error in invoke RESTDyCNorma.detectaErrores: " + ex);  
 }  
 return dom;  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/corrigeErrores", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioDyCNorma corrigeErrores(@RequestBody DominioDyCNorma dom) throws Exception {  
 dom.setStatus(false);  
 dom.setCode(500);  
 try {  
 LOG.info("invoke RESTDyCNorma.corrigeErrores. . .");  
 dom = businessIfaceCorreccionNorma.corrigeErrores(dom);  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 LOG.error("Error in invoke RESTDyCNorma.corrigeErrores: " + ex);  
 dom.setMsg("Error in invoke RESTDyCNorma.corrigeErrores: " + ex);  
 }  
 return dom;  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/getTags", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioReporteDyC getTags(@RequestBody DominioDyCNorma dom) throws Exception {  
 DominioReporteDyC tags = new DominioReporteDyC();  
 tags.setStatus(false);  
 tags.setCode(500);  
 try {  
 LOG.info("invoke RESTDyCNorma.getTags. . .");  
 tags = businessIfaceDyCNorma.getTags(dom);  
 LOG.info("####### tags size(): " + tags.getTags().size());  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 LOG.error("Error in invoke RESTDyCNorma.getTags: " + ex);  
 dom.setMsg("Error in invoke RESTDyCNorma.getTags: " + ex);  
 }  
 return tags;  
 }  
  
 private HashTags mockData() {  
 HashTags tags = new HashTags();  
 List<TagKey> listaTagKeys = new ArrayList<>();  
 TagKey tagKey1 = new TagKey();  
 tagKey1.setTagName("TAMB101");  
 List<TagValue> listaTagValues = new ArrayList<>();  
 TagValue tagVal1 = new TagValue();  
 tagVal1.setTagKey("key 1");  
 List<String> valores = new ArrayList<>();  
 valores.add("v1");  
 valores.add("v2");  
 valores.add("v3");  
 tagVal1.setTagValues(valores);  
 listaTagValues.add(tagVal1);  
 tagKey1.setTagData(listaTagValues);  
 listaTagKeys.add(tagKey1);  
 tags.setTags(listaTagKeys);  
 return tags;  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/getHashTags", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 HashTags getHasTags(@RequestBody DominioReporteDyC tags) throws Exception {  
  
 //Cambia el formato de las fechas de MM/dd/yyyy a yyyy/MM/dd  
 String[] rangosFechas = tags.getRangosFechas()  
 .trim()  
 .split("\\-");  
 SimpleDateFormat formatSrc = new SimpleDateFormat("MM/dd/yyyy");  
 SimpleDateFormat formatDst = new SimpleDateFormat("yyyy/MM/dd");  
 Date dateInit = formatSrc.parse(rangosFechas[0]);  
 Date dateFinal = formatSrc.parse(rangosFechas[1]);  
 String fechaInicial = formatDst.format(dateInit);  
 String fechaFinal = formatDst.format(dateFinal);  
   
 /\*  
 String fechaInicial = tags.getRangosFechas().split("-")[0];  
 String fechaFinal = tags.getRangosFechas().split("-")[1];  
 System.out.println("desde:" + fechaInicial + " hasta:" + fechaFinal);\*/  
 HashTags hashTags = new HashTags();  
 List<TagKey> tagKeyList = new ArrayList<>();  
 TagKey tagKeyPadre = null;  
 //Itera a traves de cada uno de los tags seleccionados  
 for (DominioCatTags tag : tags.getTags()) {  
 TagKey tagKey = new TagKey();  
 // System.out.println("Tag id a buscar:" + tag.getTagId());  
 //Obtiene el tag de la base de datos  
 DominioCatTags tagTotal = (DominioCatTags) businessCatTags.findById(tag);  
 // System.out.println("query:" + tagTotal.getQueryInformacion());  
 if (tagTotal.getQueryInformacion() != null) {  
 CatTags tagEntity = new CatTags();  
 JRUtil.copyProperties(tagEntity, tagTotal);  
 // System.out.println("query en entity:" + tagEntity.getQueryInformacion());  
 List<TagValue> tagValuesList = tagsInterface.ejecutaQuery(tagEntity, fechaInicial, fechaFinal);  
 tagKey.setTagId(tag.getTagId());  
 if (tagTotal.getPadreId() != null) {  
 tagKeyPadre = new TagKey();  
 tagKeyPadre.setTagId(tagTotal.getPadreId().getTagId());  
 tagKeyPadre.setTagName(tagTotal.getPadreId().getTag());  
 tagKeyPadre.setTagDescripcion(tagTotal.getPadreId().getTagDescripcion());  
 tagKey.setPadreId(tagKeyPadre);  
 }  
 tagKey.setTagName(tagTotal.getTag());  
 tagKey.setTagData(tagValuesList);  
  
 if (tagTotal.getInformacionAdicional() != null) {  
 List<String> informacion = new ArrayList<>();  
  
 StringTokenizer cadenas = new StringTokenizer(tagTotal.getInformacionAdicional(), "|");  
 while (cadenas.hasMoreTokens()) {  
 informacion.add(cadenas.nextToken());  
 }  
 tagKey.setInformacionAdicional(informacion);  
 }  
  
  
 tagKey.setTagDescripcion(tagTotal.getTagDescripcion());  
 tagKeyList.add(tagKey);  
 }  
 }  
 hashTags.setTags(tagKeyList);  
 return hashTags;  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/getValoresTags", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioReporteDyC getValoresTags(@RequestBody DominioReporteDyC tags) throws Exception {  
 System.out.println("################################ Tags Values: " + tags.getValuesTags());  
 DominioReporteDyC reporte = new DominioReporteDyC();  
 reporte.setStatus(false);  
 reporte.setCode(500);  
 Map<String, List<String>> map = new HashMap<>();  
 for (int key = 1; key < 13; key++) {  
 if (key != 9 && key != 11) {  
 map.put("" + key, new ArrayList<>());  
 }  
 }  
 try {  
 String tagsValues = tags.getValuesTags();  
 tagsValues = tagsValues.substring(0, tagsValues.length() - 1);  
 //System.out.println("######## Values Tags: "+tagsValues);  
 //System.out.println("######## Planta Id from REST: "+tags.getPlanta().getPlantaId());  
 String[] valuesTags = tags.getValuesTags().split("\\|");  
 for (String val : valuesTags) {  
 if (val.contains("\_")) {  
 String[] idVarStatus = val.split("\\\_");  
 String key = idVarStatus[0];  
 String status = idVarStatus[1];  
 map.get(key).add(status);  
 //System.out.println("####### Mapa add key: "+key+" - Dato: "+status+" Size mapa: "+map.get(key).size());  
 }  
 }  
 for (String key : map.keySet()) {  
 DominioTags tag = new DominioTags();  
 DominioCatVariable variable = new DominioCatVariable();  
 variable.setVariableId(new BigDecimal(key));  
 tag.setVariable(variable);  
 List<String> estatuses = map.get(key);  
 //System.out.println("####### estatuses mapa size() key - Var id: "+key+" - "+estatuses.size());  
 if (estatuses != null && estatuses.size() > 0) {  
 for (String status : estatuses) {  
 DominioCatEstatusVariable varStatus = new DominioCatEstatusVariable();  
 varStatus.setEstatusVarId(new BigDecimal(status));  
 tag.getEstatuses().add(varStatus);  
 }  
 // tags.getTags().add(tag);  
 }  
 }  
 if (tags.getRangosFechas() != null && tags.getRangosFechas().length() > 0) {  
 String[] rangosFechas = tags.getRangosFechas()  
 .trim()  
 .split("\\-");  
 DateFormat df = new SimpleDateFormat("MM/dd/yyyy");  
 Date date = new Date();  
 date = df.parse(rangosFechas[0]);  
 tags.setFechaInicio(date);  
 date = df.parse(rangosFechas[1]);  
 tags.setFechaFin(date);  
 }  
 System.out.println("##### Fecha de Inicio: " + tags.getFechaInicio());  
 System.out.println("##### Fecha de Fin: " + tags.getFechaFin());  
 LOG.info("invoke RESTDyCNorma.getTags. . .");  
 reporte = businessIfaceDyCNorma.getValoresTags(tags);  
 System.out.println("####### tags size(): " + reporte.getTags().size());  
 LOG.info("####### tags size(): " + reporte.getTags().size());  
 } catch (Exception ex) {  
  
 ex.printStackTrace();  
 LOG.error("Error in invoke RESTDyCNorma.getTags: " + ex);  
 reporte.setMsg("Error in invoke RESTDyCNorma.getTags: " + ex);  
 }  
 return reporte;  
 }  
  
 /\*\* nueva version de generaExcel que permite exportar Formato, Norma y Perfiles juntos  
 \*  
 \* @param dom contiene la informacion necesaria como rango de fechas y tipo(s) de informacion a incluir  
 \*/  
 @RequestMapping(method = RequestMethod.POST, path = "/generaExcel", consumes = "application/json", produces = "application/json")  
 public @ResponseBody  
 DominioDyCNorma generaExcel(@RequestBody DominioDyCNorma dom) throws Exception {  
 parsearRangoDeFechas(dom);  
 try {  
 System.out.println("invoke RESTDyCNorma.generaExcel. . .");  
 //Spring mapea y llama a BusinessDeteccionNorma.generaExcel(dom)  
 businessIfaceDeteccionNorma.generaExcel(dom);  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 System.out.println("Error en RESTDyCNorma.generaExcelFormato al invocar generaExcel(): " + ex);  
 dom.setMsg("Error en RESTDyCNorma.generaExcelFormato al invocar generaExcel(): " + ex);  
 }  
 return dom;  
 }  
  
 /\*\* llamado directamente por dycNorma.js, con un String que representa el rango de fechas de los cincominutales  
 \* a seleccionar.  
 \* @param dom inicializado por spring usando rangosFechas de dycNorma.js\*/  
 @RequestMapping(method = RequestMethod.POST, path = "/generaExcelDF", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioDyCNorma generaExcelFormato(@RequestBody DominioDyCNorma dom) throws Exception {  
 //Parsea y da formato a la fecha enviada por dycNorma.js  
 parsearRangoDeFechas(dom);  
 try {  
 System.out.println("invoke RESTDyCNorma.generaExcel. . .");  
 //Spring mapea y llama a BusinessDeteccionNorma.generaExcel(dom, String)  
 businessIfaceDeteccionNorma.generaExcel(dom, "Formato");  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 System.out.println("Error en RESTDyCNorma.generaExcelFormato al invocar generaExcel(): " + ex);  
 dom.setMsg("Error en RESTDyCNorma.generaExcelFormato al invocar generaExcel(): " + ex);  
 }  
 return dom;  
 } //FIN: generaExcelDN()  
  
 /\*\* llamado directamente por dycNorma.js, con un String que representa el rango de fechas de los cincominutales  
 \* a seleccionar.  
 \* @param dom inicializado por spring usando rangosFechas de dycNorma.js\*/  
 @RequestMapping(method = RequestMethod.POST, path = "/generaExcelDN", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioDyCNorma generaExcelNorma(@RequestBody DominioDyCNorma dom) throws Exception {  
 //Parsea y da formato a la fecha enviada por dycNorma.js  
 parsearRangoDeFechas(dom);  
 try {  
 System.out.println("invoke RESTDyCNorma.generaExcel. . .");  
 //Spring mapea y llama a BusinessDeteccionNorma.generaExcel(dom, String)  
 businessIfaceDeteccionNorma.generaExcel(dom, "Norma");  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 System.out.println("Error en RESTDyCNorma.generaExcelNorma al invocar generaExcel(): " + ex);  
 dom.setMsg("Error en RESTDyCNorma.generaExcelNorma al invocar generaExcel(): " + ex);  
 }  
 return dom;  
 } //FIN: generaExcelDN()  
  
  
 /\*\* llamado directamente por dycNorma.js, con un String que representa el rango de fechas de los cincominutales  
 \* a seleccionar.  
 \* @param dom inicializado por spring usando rangosFechas de dycNorma.js\*/  
 @RequestMapping(method = RequestMethod.POST, path = "/generaExcelPerfiles", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioDyCNorma generaExcelPerfiles(@RequestBody DominioDyCNorma dom) throws Exception {  
 //Parsea y da formato a la fecha enviada por dycNorma.js  
 parsearRangoDeFechas(dom);  
 try {  
 System.out.println("invoke RESTDyCNorma.generaExcelPerfiles. . .");  
 //Spring mapea y llama a BusinessDeteccionNorma.generaExcel(dom, String)  
 businessIfaceDeteccionNorma.generaExcel(dom, "Perfiles");  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 System.out.println("Error en RESTDyCNorma.generaExcelPerfiles al invocar generaExcel(): " + ex);  
 dom.setMsg("Error en RESTDyCNorma.generaExcelPerfiles al invocar generaExcel(): " + ex);  
 }  
 return dom;  
 } //FIN: generaExcelPerfiles()  
  
 private void parsearRangoDeFechas(@RequestBody DominioDyCNorma dom) {  
 String[] rangosFechas = dom.getRangosFechas()  
 .trim()  
 .split("\\-");  
  
 dom.setFechaInicio(rangosFechas[0]);  
 dom.setFechaFin(rangosFechas[1]);  
 dom.setStatus(false);  
 dom.setCode(500);  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\negocio\RESTDyCNormaPerfiles.java

package mx.mbi.dgc.microservices.controller.negocio;  
  
import java.math.BigDecimal;  
  
import java.text.DateFormat;  
import java.text.SimpleDateFormat;  
  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import mx.mbi.dgc.business.dyc.IfaceCorreccionNormaPerfiles;  
import mx.mbi.dgc.business.dyc.IfaceDeteccionNormaPerfiles;  
import mx.mbi.dgc.business.dyc.IfaceDyCNorma;  
import mx.mbi.dgc.dominio.DominioCatEstatusVariable;  
import mx.mbi.dgc.dominio.DominioCatVariable;  
import mx.mbi.dgc.dominio.DominioDyCNorma;  
import mx.mbi.dgc.dominio.DominioReporteDyC;  
import mx.mbi.dgc.dominio.DominioTags;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/DyCNormaPerfiles")  
public class RESTDyCNormaPerfiles {  
 private static final Logger LOG = LogManager.getLogger(RESTDyCNormaPerfiles.class);  
 @Autowired  
 private IfaceDeteccionNormaPerfiles businessIfaceDeteccionNormaPerfiles;  
  
 @Autowired  
 private IfaceCorreccionNormaPerfiles businessIfaceCorreccionNormaPerfiles;  
  
 @Autowired  
 private IfaceDyCNorma businessIfaceDyCNorma;  
  
 @RequestMapping(method = RequestMethod.POST, path = "/detectaErrores", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioDyCNorma detectaErrores(@RequestBody DominioDyCNorma dom) throws Exception {  
 dom.setStatus(false);  
 dom.setCode(500);  
 try {  
 LOG.info("invoke RESTDyCNorma.detectaErrores. . .");  
 dom = businessIfaceDeteccionNormaPerfiles.detectaErrores(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTDyCNorma.detectaErrores: " + ex);  
 dom.setMsg("Error in invoke RESTDyCNorma.detectaErrores: " + ex);  
 }  
 return dom;  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/corrigeErrores", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioDyCNorma corrigeErrores(@RequestBody DominioDyCNorma dom) throws Exception {  
 dom.setStatus(false);  
 dom.setCode(500);  
 try {  
 System.out.println("invoke RESTDyCNorma.corrigeErrores. . .");  
 dom = businessIfaceCorreccionNormaPerfiles.corrigeErrores(dom);  
 } catch (Exception ex) {  
 System.out.println("Error in invoke RESTDyCNorma.corrigeErrores: " + ex);  
 dom.setMsg("Error in invoke RESTDyCNorma.corrigeErrores: " + ex);  
 }  
 return dom;  
 }  
   
 @RequestMapping(method = RequestMethod.POST, path = "/generaExcel", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioDyCNorma generaExcel(@RequestBody DominioDyCNorma dom) throws Exception {  
 return dom;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\negocio\RESTLogin.java

package mx.mbi.dgc.microservices.controller.negocio;  
  
import mx.mbi.dgc.business.IfaceLogin;  
import mx.mbi.dgc.dominio.DominioTxUsuario;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/login")  
public class RESTLogin {  
 private static final Logger LOG = LogManager.getLogger(RESTLogin.class);  
 @Autowired  
 private IfaceLogin businessTxUsuario;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findByUsername",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuario findByUsername(@RequestBody DominioTxUsuario dom) throws Exception {  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setFound(false);  
 try {  
 LOG.info("invoke RESTTxUsuario.findByUsername. . .");  
 return businessTxUsuario.findByUsername(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuario.findByUsername: "+ex);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatAnioMes.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatAnioMes;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatAnioMes;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatAnioMes")  
public class RESTCatAnioMes {  
 private static final Logger LOG = LogManager.getLogger(RESTCatAnioMes.class);  
 @Autowired  
 private IfaceCatAnioMes businessCatAnioMes;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatAnioMes> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatAnioMes.findAll. . .");  
 List<DominioCatAnioMes> lista = businessCatAnioMes.findAll();  
 LOG.info("invoke RESTCatAnioMes.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatAnioMes.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatAnioMes dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatAnioMes.findAllPagination. . .");  
 response = businessCatAnioMes.allPagination(dom);  
 LOG.info("invoke RESTCatAnioMes.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatAnioMes.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatAnioMes.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatAnioMes save(@RequestBody DominioCatAnioMes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatAnioMes.save. . .");  
 LOG.info("dom.getAnioMesId(): "+dom.getAnioMesId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatAnioMes)businessCatAnioMes.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatAnioMes.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatAnioMes update(@RequestBody DominioCatAnioMes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatAnioMes.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatAnioMes)businessCatAnioMes.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatAnioMes.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatAnioMes delete(@RequestBody DominioCatAnioMes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatAnioMes.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatAnioMes)businessCatAnioMes.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatAnioMes.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatAnioMes findById(@RequestBody DominioCatAnioMes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatAnioMes.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatAnioMes)businessCatAnioMes.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatAnioMes.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatAnioMes count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatAnioMes.count. . .");  
 DominioCatAnioMes dom = new DominioCatAnioMes();  
 dom.setCount(businessCatAnioMes.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatAnioMes.count: "+ex);  
 DominioCatAnioMes dom = new DominioCatAnioMes();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatAnioMes existsById(@RequestBody DominioCatAnioMes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatAnioMes.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatAnioMes)businessCatAnioMes.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatAnioMes.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatCapacidad.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatCapacidad;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCapacidad;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatCapacidad")  
public class RESTCatCapacidad {  
 private static final Logger LOG = LogManager.getLogger(RESTCatCapacidad.class);  
 @Autowired  
 private IfaceCatCapacidad businessCatCapacidad;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatCapacidad> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCapacidad.findAll. . .");  
 List<DominioCatCapacidad> lista = businessCatCapacidad.findAll();  
 LOG.info("invoke RESTCatCapacidad.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCapacidad.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatCapacidad dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatCapacidad.findAllPagination. . .");  
 response = businessCatCapacidad.allPagination(dom);  
 LOG.info("invoke RESTCatCapacidad.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCapacidad.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatCapacidad.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCapacidad save(@RequestBody DominioCatCapacidad dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCapacidad.save. . .");  
 LOG.info("dom.getCapacidadId(): "+dom.getCapacidadId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCapacidad)businessCatCapacidad.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCapacidad.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCapacidad update(@RequestBody DominioCatCapacidad dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCapacidad.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCapacidad)businessCatCapacidad.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCapacidad.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCapacidad delete(@RequestBody DominioCatCapacidad dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCapacidad.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCapacidad)businessCatCapacidad.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCapacidad.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCapacidad findById(@RequestBody DominioCatCapacidad dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCapacidad.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCapacidad)businessCatCapacidad.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCapacidad.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCapacidad count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCapacidad.count. . .");  
 DominioCatCapacidad dom = new DominioCatCapacidad();  
 dom.setCount(businessCatCapacidad.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCapacidad.count: "+ex);  
 DominioCatCapacidad dom = new DominioCatCapacidad();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCapacidad existsById(@RequestBody DominioCatCapacidad dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCapacidad.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatCapacidad)businessCatCapacidad.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCapacidad.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatCargoFijo.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatCargoFijo;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCargoFijo;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatCargoFijo")  
public class RESTCatCargoFijo {  
 private static final Logger LOG = LogManager.getLogger(RESTCatCargoFijo.class);  
 @Autowired  
 private IfaceCatCargoFijo businessCatCargoFijo;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatCargoFijo> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCargoFijo.findAll. . .");  
 List<DominioCatCargoFijo> lista = businessCatCargoFijo.findAll();  
 LOG.info("invoke RESTCatCargoFijo.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargoFijo.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatCargoFijo dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatCargoFijo.findAllPagination. . .");  
 response = businessCatCargoFijo.allPagination(dom);  
 LOG.info("invoke RESTCatCargoFijo.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargoFijo.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatCargoFijo.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCargoFijo save(@RequestBody DominioCatCargoFijo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCargoFijo.save. . .");  
 LOG.info("dom.getCargoFijoId(): "+dom.getCargoFijoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCargoFijo)businessCatCargoFijo.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargoFijo.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCargoFijo update(@RequestBody DominioCatCargoFijo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCargoFijo.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCargoFijo)businessCatCargoFijo.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargoFijo.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCargoFijo delete(@RequestBody DominioCatCargoFijo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCargoFijo.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCargoFijo)businessCatCargoFijo.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargoFijo.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCargoFijo findById(@RequestBody DominioCatCargoFijo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCargoFijo.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCargoFijo)businessCatCargoFijo.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargoFijo.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCargoFijo count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCargoFijo.count. . .");  
 DominioCatCargoFijo dom = new DominioCatCargoFijo();  
 dom.setCount(businessCatCargoFijo.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargoFijo.count: "+ex);  
 DominioCatCargoFijo dom = new DominioCatCargoFijo();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCargoFijo existsById(@RequestBody DominioCatCargoFijo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCargoFijo.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatCargoFijo)businessCatCargoFijo.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargoFijo.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatCargos.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatCargos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCargos;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatCargos")  
public class RESTCatCargos {  
 private static final Logger LOG = LogManager.getLogger(RESTCatCargos.class);  
 @Autowired  
 private IfaceCatCargos businessCatCargos;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatCargos> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCargos.findAll. . .");  
 List<DominioCatCargos> lista = businessCatCargos.findAll();  
 LOG.info("invoke RESTCatCargos.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargos.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatCargos dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatCargos.findAllPagination. . .");  
 response = businessCatCargos.allPagination(dom);  
 LOG.info("invoke RESTCatCargos.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargos.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatCargos.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCargos save(@RequestBody DominioCatCargos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCargos.save. . .");  
 LOG.info("dom.getCargoId(): "+dom.getCargoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCargos)businessCatCargos.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargos.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCargos update(@RequestBody DominioCatCargos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCargos.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCargos)businessCatCargos.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargos.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCargos delete(@RequestBody DominioCatCargos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCargos.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCargos)businessCatCargos.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargos.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCargos findById(@RequestBody DominioCatCargos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCargos.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCargos)businessCatCargos.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargos.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCargos count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCargos.count. . .");  
 DominioCatCargos dom = new DominioCatCargos();  
 dom.setCount(businessCatCargos.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargos.count: "+ex);  
 DominioCatCargos dom = new DominioCatCargos();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCargos existsById(@RequestBody DominioCatCargos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCargos.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatCargos)businessCatCargos.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCargos.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatClasificacion.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatClasificacion;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatClasificacion;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatClasificacion")  
public class RESTCatClasificacion {  
 private static final Logger LOG = LogManager.getLogger(RESTCatClasificacion.class);  
 @Autowired  
 private IfaceCatClasificacion businessCatClasificacion;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatClasificacion> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacion.findAll. . .");  
 List<DominioCatClasificacion> lista = businessCatClasificacion.findAll();  
 LOG.info("invoke RESTCatClasificacion.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacion.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatClasificacion dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatClasificacion.findAllPagination. . .");  
 response = businessCatClasificacion.allPagination(dom);  
 LOG.info("invoke RESTCatClasificacion.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacion.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatClasificacion.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacion save(@RequestBody DominioCatClasificacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacion.save. . .");  
 LOG.info("dom.getClasificacionId(): "+dom.getClasificacionId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClasificacion)businessCatClasificacion.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacion.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacion update(@RequestBody DominioCatClasificacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacion.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClasificacion)businessCatClasificacion.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacion.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacion delete(@RequestBody DominioCatClasificacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacion.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClasificacion)businessCatClasificacion.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacion.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacion findById(@RequestBody DominioCatClasificacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacion.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClasificacion)businessCatClasificacion.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacion.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacion count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacion.count. . .");  
 DominioCatClasificacion dom = new DominioCatClasificacion();  
 dom.setCount(businessCatClasificacion.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacion.count: "+ex);  
 DominioCatClasificacion dom = new DominioCatClasificacion();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacion existsById(@RequestBody DominioCatClasificacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacion.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatClasificacion)businessCatClasificacion.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacion.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatClasificacionEvento.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceCatClasificacionEvento;  
import mx.mbi.dgc.dominio.DominioCatClasificacionEvento;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatClasificacionEvento")  
public class RESTCatClasificacionEvento {  
 private static final Logger LOG = LogManager.getLogger(RESTCatClasificacionEvento.class);  
 @Autowired  
 private IfaceCatClasificacionEvento businessCatClasificacionEvento;  
  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatClasificacionEvento> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacionEvento.findAll. . .");  
 List<DominioCatClasificacionEvento> lista = businessCatClasificacionEvento.findAll();  
 LOG.info("invoke RESTCatClasificacionEvento.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacionEvento.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatClasificacionEvento dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatClasificacionEvento.findAllPagination. . .");  
 response = businessCatClasificacionEvento.allPagination(dom);  
 LOG.info("invoke RESTCatClasificacionEvento.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 ex.printStackTrace();  
 LOG.error("Error in invoke RESTCatClasificacionEvento.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatClasificacionEvento.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacionEvento save(@RequestBody DominioCatClasificacionEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacionEvento.save. . .");  
 LOG.info("dom.getClasificacionEventoId(): "+dom.getClasificacionEventoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClasificacionEvento)businessCatClasificacionEvento.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacionEvento.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacionEvento update(@RequestBody DominioCatClasificacionEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacionEvento.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClasificacionEvento)businessCatClasificacionEvento.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacionEvento.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacionEvento delete(@RequestBody DominioCatClasificacionEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacionEvento.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClasificacionEvento)businessCatClasificacionEvento.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacionEvento.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacionEvento findById(@RequestBody DominioCatClasificacionEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacionEvento.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClasificacionEvento)businessCatClasificacionEvento.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacionEvento.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacionEvento count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacionEvento.count. . .");  
 DominioCatClasificacionEvento dom = new DominioCatClasificacionEvento();  
 dom.setCount(businessCatClasificacionEvento.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacionEvento.count: "+ex);  
 DominioCatClasificacionEvento dom = new DominioCatClasificacionEvento();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacionEvento existsById(@RequestBody DominioCatClasificacionEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacionEvento.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatClasificacionEvento)businessCatClasificacionEvento.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacionEvento.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatClasificacionEventos.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceCatClasificacionEventos;  
import mx.mbi.dgc.dominio.DominioCatClasificacionEventos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatClasificacionEventos")  
public class RESTCatClasificacionEventos {  
 private static final Logger LOG = LogManager.getLogger(RESTCatClasificacionEventos.class);  
 @Autowired  
 private IfaceCatClasificacionEventos businessCatClasificacionEventos;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatClasificacionEventos> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacionEventos.findAll. . .");  
 List<DominioCatClasificacionEventos> lista = businessCatClasificacionEventos.findAll();  
 LOG.info("invoke RESTCatClasificacionEventos.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacionEventos.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatClasificacionEventos dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatClasificacionEventos.findAllPagination. . .");  
 response = businessCatClasificacionEventos.allPagination(dom);  
 LOG.info("invoke RESTCatClasificacionEventos.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacionEventos.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatClasificacionEventos.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacionEventos save(@RequestBody DominioCatClasificacionEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacionEventos.save. . .");  
 LOG.info("dom.getClasificacionEventoId(): "+dom.getClasificacionEventoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClasificacionEventos)businessCatClasificacionEventos.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacionEventos.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacionEventos update(@RequestBody DominioCatClasificacionEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacionEventos.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClasificacionEventos)businessCatClasificacionEventos.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacionEventos.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacionEventos delete(@RequestBody DominioCatClasificacionEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacionEventos.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClasificacionEventos)businessCatClasificacionEventos.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacionEventos.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacionEventos findById(@RequestBody DominioCatClasificacionEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacionEventos.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClasificacionEventos)businessCatClasificacionEventos.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacionEventos.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacionEventos count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacionEventos.count. . .");  
 DominioCatClasificacionEventos dom = new DominioCatClasificacionEventos();  
 dom.setCount(businessCatClasificacionEventos.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacionEventos.count: "+ex);  
 DominioCatClasificacionEventos dom = new DominioCatClasificacionEventos();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClasificacionEventos existsById(@RequestBody DominioCatClasificacionEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClasificacionEventos.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatClasificacionEventos)businessCatClasificacionEventos.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClasificacionEventos.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatClienteMonedas.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatClienteMonedas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatClienteMonedas;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatClienteMonedas")  
public class RESTCatClienteMonedas {  
 private static final Logger LOG = LogManager.getLogger(RESTCatClienteMonedas.class);  
 @Autowired  
 private IfaceCatClienteMonedas businessCatClienteMonedas;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatClienteMonedas> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatClienteMonedas.findAll. . .");  
 List<DominioCatClienteMonedas> lista = businessCatClienteMonedas.findAll();  
 LOG.info("invoke RESTCatClienteMonedas.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClienteMonedas.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatClienteMonedas dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatClienteMonedas.findAllPagination. . .");  
 response = businessCatClienteMonedas.allPagination(dom);  
 LOG.info("invoke RESTCatClienteMonedas.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClienteMonedas.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatClienteMonedas.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClienteMonedas save(@RequestBody DominioCatClienteMonedas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClienteMonedas.save. . .");  
 LOG.info("dom.getClienteMonedaId(): "+dom.getClienteMonedaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClienteMonedas)businessCatClienteMonedas.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClienteMonedas.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClienteMonedas update(@RequestBody DominioCatClienteMonedas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClienteMonedas.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClienteMonedas)businessCatClienteMonedas.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClienteMonedas.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClienteMonedas delete(@RequestBody DominioCatClienteMonedas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClienteMonedas.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClienteMonedas)businessCatClienteMonedas.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClienteMonedas.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClienteMonedas findById(@RequestBody DominioCatClienteMonedas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClienteMonedas.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClienteMonedas)businessCatClienteMonedas.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClienteMonedas.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClienteMonedas count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatClienteMonedas.count. . .");  
 DominioCatClienteMonedas dom = new DominioCatClienteMonedas();  
 dom.setCount(businessCatClienteMonedas.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClienteMonedas.count: "+ex);  
 DominioCatClienteMonedas dom = new DominioCatClienteMonedas();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClienteMonedas existsById(@RequestBody DominioCatClienteMonedas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClienteMonedas.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatClienteMonedas)businessCatClienteMonedas.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClienteMonedas.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatClientes.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatClientes;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatClientes;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatClientes")  
public class RESTCatClientes {  
 private static final Logger LOG = LogManager.getLogger(RESTCatClientes.class);  
 @Autowired  
 private IfaceCatClientes businessCatClientes;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatClientes> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatClientes.findAll. . .");  
 List<DominioCatClientes> lista = businessCatClientes.findAll();  
 LOG.info("invoke RESTCatClientes.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClientes.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatClientes dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatClientes.findAllPagination. . .");  
 response = businessCatClientes.allPagination(dom);  
 LOG.info("invoke RESTCatClientes.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClientes.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatClientes.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClientes save(@RequestBody DominioCatClientes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClientes.save. . .");  
 LOG.info("dom.getClienteId(): "+dom.getClienteId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClientes)businessCatClientes.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClientes.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClientes update(@RequestBody DominioCatClientes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClientes.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClientes)businessCatClientes.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClientes.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClientes delete(@RequestBody DominioCatClientes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClientes.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClientes)businessCatClientes.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClientes.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClientes findById(@RequestBody DominioCatClientes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClientes.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatClientes)businessCatClientes.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClientes.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClientes count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatClientes.count. . .");  
 DominioCatClientes dom = new DominioCatClientes();  
 dom.setCount(businessCatClientes.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClientes.count: "+ex);  
 DominioCatClientes dom = new DominioCatClientes();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatClientes existsById(@RequestBody DominioCatClientes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatClientes.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatClientes)businessCatClientes.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatClientes.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatCombustible.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatCombustible;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCombustible;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatCombustible")  
public class RESTCatCombustible {  
 private static final Logger LOG = LogManager.getLogger(RESTCatCombustible.class);  
 @Autowired  
 private IfaceCatCombustible businessCatCombustible;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatCombustible> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCombustible.findAll. . .");  
 List<DominioCatCombustible> lista = businessCatCombustible.findAll();  
 LOG.info("invoke RESTCatCombustible.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCombustible.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatCombustible dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatCombustible.findAllPagination. . .");  
 response = businessCatCombustible.allPagination(dom);  
 LOG.info("invoke RESTCatCombustible.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCombustible.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatCombustible.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCombustible save(@RequestBody DominioCatCombustible dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCombustible.save. . .");  
 LOG.info("dom.getCombustibleId(): "+dom.getCombustibleId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCombustible)businessCatCombustible.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCombustible.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCombustible update(@RequestBody DominioCatCombustible dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCombustible.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCombustible)businessCatCombustible.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCombustible.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCombustible delete(@RequestBody DominioCatCombustible dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCombustible.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCombustible)businessCatCombustible.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCombustible.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCombustible findById(@RequestBody DominioCatCombustible dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCombustible.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCombustible)businessCatCombustible.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCombustible.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCombustible count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCombustible.count. . .");  
 DominioCatCombustible dom = new DominioCatCombustible();  
 dom.setCount(businessCatCombustible.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCombustible.count: "+ex);  
 DominioCatCombustible dom = new DominioCatCombustible();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCombustible existsById(@RequestBody DominioCatCombustible dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCombustible.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatCombustible)businessCatCombustible.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCombustible.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatCompania.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatCompania;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCompania;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatCompania")  
public class RESTCatCompania {  
 private static final Logger LOG = LogManager.getLogger(RESTCatCompania.class);  
 @Autowired  
 private IfaceCatCompania businessCatCompania;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatCompania> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCompania.findAll. . .");  
 List<DominioCatCompania> lista = businessCatCompania.findAll();  
 LOG.info("invoke RESTCatCompania.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompania.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatCompania dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatCompania.findAllPagination. . .");  
 response = businessCatCompania.allPagination(dom);  
 LOG.info("invoke RESTCatCompania.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompania.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatCompania.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCompania save(@RequestBody DominioCatCompania dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCompania.save. . .");  
 LOG.info("dom.getCompaniaId(): "+dom.getCompaniaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCompania)businessCatCompania.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompania.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCompania update(@RequestBody DominioCatCompania dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCompania.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCompania)businessCatCompania.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompania.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCompania delete(@RequestBody DominioCatCompania dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCompania.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCompania)businessCatCompania.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompania.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCompania findById(@RequestBody DominioCatCompania dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCompania.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCompania)businessCatCompania.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompania.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCompania count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCompania.count. . .");  
 DominioCatCompania dom = new DominioCatCompania();  
 dom.setCount(businessCatCompania.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompania.count: "+ex);  
 DominioCatCompania dom = new DominioCatCompania();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCompania existsById(@RequestBody DominioCatCompania dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCompania.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatCompania)businessCatCompania.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompania.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatCompaniaPlantas.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatCompaniaPlantas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCompaniaPlantas;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatCompaniaPlantas")  
public class RESTCatCompaniaPlantas {  
 private static final Logger LOG = LogManager.getLogger(RESTCatCompaniaPlantas.class);  
 @Autowired  
 private IfaceCatCompaniaPlantas businessCatCompaniaPlantas;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatCompaniaPlantas> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCompaniaPlantas.findAll. . .");  
 List<DominioCatCompaniaPlantas> lista = businessCatCompaniaPlantas.findAll();  
 LOG.info("invoke RESTCatCompaniaPlantas.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompaniaPlantas.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatCompaniaPlantas dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatCompaniaPlantas.findAllPagination. . .");  
 response = businessCatCompaniaPlantas.allPagination(dom);  
 LOG.info("invoke RESTCatCompaniaPlantas.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompaniaPlantas.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatCompaniaPlantas.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCompaniaPlantas save(@RequestBody DominioCatCompaniaPlantas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCompaniaPlantas.save. . .");  
 LOG.info("dom.getCompaniaPlantaId(): "+dom.getCompaniaPlantaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCompaniaPlantas)businessCatCompaniaPlantas.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompaniaPlantas.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCompaniaPlantas update(@RequestBody DominioCatCompaniaPlantas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCompaniaPlantas.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCompaniaPlantas)businessCatCompaniaPlantas.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompaniaPlantas.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCompaniaPlantas delete(@RequestBody DominioCatCompaniaPlantas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCompaniaPlantas.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCompaniaPlantas)businessCatCompaniaPlantas.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompaniaPlantas.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCompaniaPlantas findById(@RequestBody DominioCatCompaniaPlantas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCompaniaPlantas.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCompaniaPlantas)businessCatCompaniaPlantas.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompaniaPlantas.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCompaniaPlantas count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCompaniaPlantas.count. . .");  
 DominioCatCompaniaPlantas dom = new DominioCatCompaniaPlantas();  
 dom.setCount(businessCatCompaniaPlantas.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompaniaPlantas.count: "+ex);  
 DominioCatCompaniaPlantas dom = new DominioCatCompaniaPlantas();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCompaniaPlantas existsById(@RequestBody DominioCatCompaniaPlantas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCompaniaPlantas.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatCompaniaPlantas)businessCatCompaniaPlantas.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCompaniaPlantas.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatComponentePeriodo.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatComponentePeriodo;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatComponentePeriodo;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatComponentePeriodo")  
public class RESTCatComponentePeriodo {  
 private static final Logger LOG = LogManager.getLogger(RESTCatComponentePeriodo.class);  
 @Autowired  
 private IfaceCatComponentePeriodo businessCatComponentePeriodo;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatComponentePeriodo> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatComponentePeriodo.findAll. . .");  
 List<DominioCatComponentePeriodo> lista = businessCatComponentePeriodo.findAll();  
 LOG.info("invoke RESTCatComponentePeriodo.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentePeriodo.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatComponentePeriodo dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatComponentePeriodo.findAllPagination. . .");  
 response = businessCatComponentePeriodo.allPagination(dom);  
 LOG.info("invoke RESTCatComponentePeriodo.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentePeriodo.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatComponentePeriodo.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatComponentePeriodo save(@RequestBody DominioCatComponentePeriodo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatComponentePeriodo.save. . .");  
 LOG.info("dom.getComponentePeriodoId(): "+dom.getComponentePeriodoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatComponentePeriodo)businessCatComponentePeriodo.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentePeriodo.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatComponentePeriodo update(@RequestBody DominioCatComponentePeriodo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatComponentePeriodo.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatComponentePeriodo)businessCatComponentePeriodo.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentePeriodo.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatComponentePeriodo delete(@RequestBody DominioCatComponentePeriodo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatComponentePeriodo.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatComponentePeriodo)businessCatComponentePeriodo.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentePeriodo.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatComponentePeriodo findById(@RequestBody DominioCatComponentePeriodo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatComponentePeriodo.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatComponentePeriodo)businessCatComponentePeriodo.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentePeriodo.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatComponentePeriodo count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatComponentePeriodo.count. . .");  
 DominioCatComponentePeriodo dom = new DominioCatComponentePeriodo();  
 dom.setCount(businessCatComponentePeriodo.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentePeriodo.count: "+ex);  
 DominioCatComponentePeriodo dom = new DominioCatComponentePeriodo();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatComponentePeriodo existsById(@RequestBody DominioCatComponentePeriodo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatComponentePeriodo.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatComponentePeriodo)businessCatComponentePeriodo.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentePeriodo.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatComponentes.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatComponentes;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatComponentes;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatComponentes")  
public class RESTCatComponentes {  
 private static final Logger LOG = LogManager.getLogger(RESTCatComponentes.class);  
 @Autowired  
 private IfaceCatComponentes businessCatComponentes;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatComponentes> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatComponentes.findAll. . .");  
 List<DominioCatComponentes> lista = businessCatComponentes.findAll();  
 LOG.info("invoke RESTCatComponentes.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentes.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatComponentes dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatComponentes.findAllPagination. . .");  
 response = businessCatComponentes.allPagination(dom);  
 LOG.info("invoke RESTCatComponentes.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentes.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatComponentes.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatComponentes save(@RequestBody DominioCatComponentes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatComponentes.save. . .");  
 LOG.info("dom.getComponenteId(): "+dom.getComponenteId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatComponentes)businessCatComponentes.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentes.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatComponentes update(@RequestBody DominioCatComponentes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatComponentes.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatComponentes)businessCatComponentes.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentes.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatComponentes delete(@RequestBody DominioCatComponentes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatComponentes.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatComponentes)businessCatComponentes.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentes.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatComponentes findById(@RequestBody DominioCatComponentes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatComponentes.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatComponentes)businessCatComponentes.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentes.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatComponentes count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatComponentes.count. . .");  
 DominioCatComponentes dom = new DominioCatComponentes();  
 dom.setCount(businessCatComponentes.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentes.count: "+ex);  
 DominioCatComponentes dom = new DominioCatComponentes();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatComponentes existsById(@RequestBody DominioCatComponentes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatComponentes.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatComponentes)businessCatComponentes.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatComponentes.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatCondiciones.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatCondiciones;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCondiciones;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatCondiciones")  
public class RESTCatCondiciones {  
 private static final Logger LOG = LogManager.getLogger(RESTCatCondiciones.class);  
 @Autowired  
 private IfaceCatCondiciones businessCatCondiciones;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatCondiciones> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCondiciones.findAll. . .");  
 List<DominioCatCondiciones> lista = businessCatCondiciones.findAll();  
 LOG.info("invoke RESTCatCondiciones.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCondiciones.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatCondiciones dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatCondiciones.findAllPagination. . .");  
 response = businessCatCondiciones.allPagination(dom);  
 LOG.info("invoke RESTCatCondiciones.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCondiciones.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatCondiciones.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCondiciones save(@RequestBody DominioCatCondiciones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCondiciones.save. . .");  
 LOG.info("dom.getCondicionId(): "+dom.getCondicionId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCondiciones)businessCatCondiciones.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCondiciones.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCondiciones update(@RequestBody DominioCatCondiciones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCondiciones.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCondiciones)businessCatCondiciones.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCondiciones.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCondiciones delete(@RequestBody DominioCatCondiciones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCondiciones.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCondiciones)businessCatCondiciones.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCondiciones.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCondiciones findById(@RequestBody DominioCatCondiciones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCondiciones.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCondiciones)businessCatCondiciones.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCondiciones.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCondiciones count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCondiciones.count. . .");  
 DominioCatCondiciones dom = new DominioCatCondiciones();  
 dom.setCount(businessCatCondiciones.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCondiciones.count: "+ex);  
 DominioCatCondiciones dom = new DominioCatCondiciones();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCondiciones existsById(@RequestBody DominioCatCondiciones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCondiciones.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatCondiciones)businessCatCondiciones.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCondiciones.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatConexiones.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatConexiones;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatConexiones;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatConexiones")  
public class RESTCatConexiones {  
 private static final Logger LOG = LogManager.getLogger(RESTCatConexiones.class);  
 @Autowired  
 private IfaceCatConexiones businessCatConexiones;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatConexiones> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatConexiones.findAll. . .");  
 List<DominioCatConexiones> lista = businessCatConexiones.findAll();  
 LOG.info("invoke RESTCatConexiones.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatConexiones.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatConexiones dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatConexiones.findAllPagination. . .");  
 response = businessCatConexiones.allPagination(dom);  
 LOG.info("invoke RESTCatConexiones.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatConexiones.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatConexiones.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatConexiones save(@RequestBody DominioCatConexiones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConexiones.save. . .");  
 LOG.info("dom.getConexionId(): "+dom.getConexionId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConexiones)businessCatConexiones.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatConexiones.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatConexiones update(@RequestBody DominioCatConexiones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConexiones.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConexiones)businessCatConexiones.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatConexiones.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatConexiones delete(@RequestBody DominioCatConexiones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConexiones.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConexiones)businessCatConexiones.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatConexiones.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatConexiones findById(@RequestBody DominioCatConexiones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConexiones.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConexiones)businessCatConexiones.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatConexiones.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatConexiones count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatConexiones.count. . .");  
 DominioCatConexiones dom = new DominioCatConexiones();  
 dom.setCount(businessCatConexiones.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatConexiones.count: "+ex);  
 DominioCatConexiones dom = new DominioCatConexiones();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatConexiones existsById(@RequestBody DominioCatConexiones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConexiones.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatConexiones)businessCatConexiones.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatConexiones.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatConfiguracionBitacora.java

package mx.mbi.dgc.microservices.controller;  
  
import java.math.BigDecimal;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceCatConfiguracionBitacora;  
import mx.mbi.dgc.business.IfaceCatConfiguracionDetalle;  
import mx.mbi.dgc.business.IfaceCatConfiguracionEventos;  
import mx.mbi.dgc.business.IfaceCatConfiguracionOpciones;  
import mx.mbi.dgc.business.IfaceCatDetalleOpciones;  
import mx.mbi.dgc.business.IfaceCatDetalles;  
import mx.mbi.dgc.dominio.DominioCatConfiguracionBitacora;  
import mx.mbi.dgc.dominio.DominioCatDetalles;  
import mx.mbi.dgc.dominio.DominioCatOpcionPlanta;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
  
import mx.mbi.dgc.business.IfaceCatOpcionPlanta;  
  
import mx.mbi.dgc.dominio.DominioCatCapacidad;  
import mx.mbi.dgc.dominio.DominioCatConfiguracionDetalle;  
import mx.mbi.dgc.dominio.DominioCatConfiguracionEventos;  
import mx.mbi.dgc.dominio.DominioCatConfiguracionOpciones;  
import mx.mbi.dgc.dominio.DominioCatDetalleOpciones;  
import mx.mbi.dgc.jpa.entity.CatConfiguracionDetalle;  
import mx.mbi.dgc.jpa.entity.CatConfiguracionEventos;  
import mx.mbi.dgc.jpa.entity.CatConfiguracionOpciones;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatConfiguracionBitacora")  
public class RESTCatConfiguracionBitacora {  
 private static final Logger LOG = LogManager.getLogger(RESTCatConfiguracionBitacora.class);  
 @Autowired  
 private IfaceCatConfiguracionEventos businessCatConfiguracionEventos;  
 @Autowired  
 private IfaceCatConfiguracionDetalle businessCatConfiguracionDetalle;  
 @Autowired  
 private IfaceCatConfiguracionOpciones businessCatConfiguracionOpciones;  
 @Autowired  
 private IfaceCatDetalles businessCatDetalles;  
 @Autowired  
 private IfaceCatDetalleOpciones businessCatDetalleOpciones;  
  
 @RequestMapping(  
 method = RequestMethod.POST, path = "/guardar",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatConfiguracionBitacora save(@RequestBody DominioCatConfiguracionBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionBitacora.save. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 long idEvento;  
 long idDetalle;  
 long idOpcion;  
 DominioCatConfiguracionEventos domConfiguracionEventos = new DominioCatConfiguracionEventos();  
 DominioCatConfiguracionDetalle domConfiguracionDetalle = new DominioCatConfiguracionDetalle();  
 JRUtil.copyProperties(domConfiguracionEventos, dom.getConfiguracionEvento());  
 idEvento = businessCatConfiguracionEventos.count();  
 idEvento = idEvento + 4;  
 domConfiguracionEventos.setConfiguracionEventoId(BigDecimal.valueOf(idEvento));  
 domConfiguracionEventos = (DominioCatConfiguracionEventos) businessCatConfiguracionEventos.save(domConfiguracionEventos);  
 for(BigDecimal configuracionOpciones : dom.getOpciones()){  
 DominioCatDetalleOpciones domDetalleOpciones = new DominioCatDetalleOpciones();  
 DominioCatDetalles domDetalles = new DominioCatDetalles();  
 System.out.println(configuracionOpciones);  
 domDetalleOpciones.setDetalleOpcionesId(configuracionOpciones);  
 domDetalleOpciones = (DominioCatDetalleOpciones) businessCatDetalleOpciones.findById(domDetalleOpciones);  
 System.out.println("id Opcion :" + domDetalleOpciones.getDetalleOpcionesId());  
 System.out.println("id Detalle en opciones :" + domDetalleOpciones.getDetalleId().getDetalleId());  
 domDetalles = (DominioCatDetalles) businessCatDetalles.findById(domDetalleOpciones.getDetalleId());  
 System.out.println("id DEtalle en detale :" + domDetalles.getDetalleId());  
 System.out.println("id configuracionEventos :" + domConfiguracionEventos.getConfiguracionEventoId());  
 List<DominioCatConfiguracionDetalle> lista1 = businessCatConfiguracionDetalle.findByCatConfiguracionEvento(domConfiguracionEventos);  
 int control = 0;  
   
 domConfiguracionDetalle.setDetalleId(domDetalles);  
 //JRUtil.copyProperties(domConfiguracionDetalle.getDetalleId(), domDetalles);  
 domConfiguracionDetalle.setConfiguracionEventoId(domConfiguracionEventos);  
 //JRUtil.copyProperties(domConfiguracionDetalle.getConfiguracionEventoId(), domConfiguracionEventos);  
 for(DominioCatConfiguracionDetalle aux : lista1)  
 if(aux.getDetalleId().getDetalleId() == domDetalles.getDetalleId())  
 control = 1;  
 System.out.println(control);  
 if(control == 0){  
 idDetalle = businessCatConfiguracionDetalle.count();  
 idDetalle = idDetalle + 6;  
 domConfiguracionDetalle.setConfiguracionDetalleId(BigDecimal.valueOf(idDetalle));  
 domConfiguracionDetalle = (DominioCatConfiguracionDetalle) businessCatConfiguracionDetalle.save(domConfiguracionDetalle);  
 }  
 DominioCatConfiguracionOpciones configuracionOpcion = new DominioCatConfiguracionOpciones();  
 configuracionOpcion.setDetalleOpcionesId(domDetalleOpciones);  
 //JRUtil.copyProperties(configuracionOpcion.getDetalleOpcionesId(), domDetalleOpciones);  
 configuracionOpcion.setConfiguracionDetalleId(domConfiguracionDetalle);  
 //JRUtil.copyProperties(configuracionOpcion.getConfiguracionDetalleId(), domConfiguracionDetalle);  
 idOpcion = businessCatConfiguracionOpciones.count();  
 idOpcion = idOpcion + 19;  
 configuracionOpcion.setConfiguracionOpcionesId(BigDecimal.valueOf(idOpcion));  
 configuracionOpcion = (DominioCatConfiguracionOpciones) businessCatConfiguracionOpciones.save(configuracionOpcion);  
 }  
 for(BigDecimal configuracionDetalles : dom.getDetalle()){  
   
 }  
 for(BigDecimal configuracionDetalles : dom.getDetalle()){  
 DominioCatDetalles domDetalles = new DominioCatDetalles();  
 domDetalles.setDetalleId(configuracionDetalles);  
 domConfiguracionDetalle.setDetalleId(domDetalles);  
 domConfiguracionDetalle.setConfiguracionEventoId(domConfiguracionEventos);  
 //JRUtil.copyProperties(configuracionDetalle.getConfiguracionDetalleId(), configuracionDetalles);  
 //JRUtil.copyProperties(configuracionDetalle.getConfiguracionEventoId(), configuracionEventos);  
 idDetalle = businessCatConfiguracionDetalle.count();  
 idDetalle = idDetalle + 6;  
 domConfiguracionDetalle.setConfiguracionDetalleId(BigDecimal.valueOf(idDetalle));  
 domConfiguracionDetalle = (DominioCatConfiguracionDetalle) businessCatConfiguracionDetalle.save(domConfiguracionDetalle);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionBitacora.guardar: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatConfiguracionDetalle.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceCatConfiguracionDetalle;  
import mx.mbi.dgc.dominio.DominioCatConfiguracionDetalle;  
import mx.mbi.dgc.dominio.DominioCatConfiguracionEventos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatConfiguracionDetalle")  
public class RESTCatConfiguracionDetalle {  
 private static final Logger LOG = LogManager.getLogger(RESTCatConfiguracionDetalle.class);  
 @Autowired  
 private IfaceCatConfiguracionDetalle businessCatConfiguracionDetalle;  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAll", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioCatConfiguracionDetalle> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionDetalle.findAll. . .");  
 List<DominioCatConfiguracionDetalle> lista = businessCatConfiguracionDetalle.findAll();  
 LOG.info("invoke RESTCatConfiguracionDetalle.findAll size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionDetalle.findAll: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findByCatConfiguracionEvento", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioCatConfiguracionDetalle> findByCatConfiguracionEvento(@RequestBody  
 DominioCatConfiguracionEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionDetalle.findByCatConfiguracionEvento. . .");  
 List<DominioCatConfiguracionDetalle> lista =  
 businessCatConfiguracionDetalle.findByCatConfiguracionEvento(dom);  
 LOG.info("invoke RESTCatConfiguracionDetalle.findAll size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionDetalle.findByCatConfiguracionEvento: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAllPagination", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioRespuesta findAllPagination(@RequestBody DominioCatConfiguracionDetalle dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatConfiguracionDetalle.findAllPagination. . .");  
 response = businessCatConfiguracionDetalle.allPagination(dom);  
 LOG.info("invoke RESTCatConfiguracionDetalle.findAllPagination size lista return: " +  
 response.getData().size());  
 return response;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionDetalle.findAllPagination: " + ex);  
 response.setMensaje("Error in invoke RESTCatConfiguracionDetalle.findAllPagination: " + ex);  
 return response;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/save", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionDetalle save(@RequestBody DominioCatConfiguracionDetalle dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionDetalle.save. . .");  
 LOG.info("dom.getConfiguracionDetalleId(): " + dom.getConfiguracionEventoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConfiguracionDetalle) businessCatConfiguracionDetalle.save(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionDetalle.save: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/update", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionDetalle update(@RequestBody DominioCatConfiguracionDetalle dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionDetalle.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConfiguracionDetalle) businessCatConfiguracionDetalle.update(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionDetalle.update: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/delete", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionDetalle delete(@RequestBody DominioCatConfiguracionDetalle dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionDetalle.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConfiguracionDetalle) businessCatConfiguracionDetalle.delete(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionDetalle.delete: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionDetalle findById(@RequestBody DominioCatConfiguracionDetalle dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionDetalle.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConfiguracionDetalle) businessCatConfiguracionDetalle.findById(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionDetalle.findById: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/count", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionDetalle count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionDetalle.count. . .");  
 DominioCatConfiguracionDetalle dom = new DominioCatConfiguracionDetalle();  
 dom.setCount(businessCatConfiguracionDetalle.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionDetalle.count: " + ex);  
 DominioCatConfiguracionDetalle dom = new DominioCatConfiguracionDetalle();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/existsById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionDetalle existsById(@RequestBody DominioCatConfiguracionDetalle dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionDetalle.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatConfiguracionDetalle) businessCatConfiguracionDetalle.existsById(dom);  
 if (dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionDetalle.existsById: " + ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatConfiguracionEventos.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceCatConfiguracionEventos;  
import mx.mbi.dgc.dominio.DominioCatClasificacionEvento;  
import mx.mbi.dgc.dominio.DominioCatConfiguracionEventos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatConfiguracionEventos")  
public class RESTCatConfiguracionEventos {  
 private static final Logger LOG = LogManager.getLogger(RESTCatConfiguracionEventos.class);  
 @Autowired  
 private IfaceCatConfiguracionEventos businessCatConfiguracionEventos;  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAll", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioCatConfiguracionEventos> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionEventos.findAll. . .");  
 List<DominioCatConfiguracionEventos> lista = businessCatConfiguracionEventos.findAll();  
 LOG.info("invoke RESTCatConfiguracionEventos.findAll size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionEventos.findAll: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findByCatClasificacionEvento", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioCatConfiguracionEventos> findByCatClasificacionEvento(@RequestBody  
 DominioCatClasificacionEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionEventos.findByCatClasificacionEvento. . .");  
 List<DominioCatConfiguracionEventos> lista =  
 businessCatConfiguracionEventos.findByCatClasificacionEvento(dom);  
 LOG.info("invoke RESTCatConfiguracionEventos.findAll size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionEventos.findByCatClasificacionEvento: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAllPagination", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioRespuesta findAllPagination(@RequestBody DominioCatConfiguracionEventos dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatConfiguracionEventos.findAllPagination. . .");  
 response = businessCatConfiguracionEventos.allPagination(dom);  
 LOG.info("invoke RESTCatConfiguracionEventos.findAllPagination size lista return: " +  
 response.getData().size());  
 return response;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionEventos.findAllPagination: " + ex);  
 response.setMensaje("Error in invoke RESTCatConfiguracionEventos.findAllPagination: " + ex);  
 return response;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/save", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionEventos save(@RequestBody DominioCatConfiguracionEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionEventos.save. . .");  
 LOG.info("dom.getConfiguracionEventosId(): " + dom.getConfiguracionEventoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConfiguracionEventos) businessCatConfiguracionEventos.save(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionEventos.save: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/update", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionEventos update(@RequestBody DominioCatConfiguracionEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionEventos.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConfiguracionEventos) businessCatConfiguracionEventos.update(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionEventos.update: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/delete", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionEventos delete(@RequestBody DominioCatConfiguracionEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionEventos.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConfiguracionEventos) businessCatConfiguracionEventos.delete(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionEventos.delete: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionEventos findById(@RequestBody DominioCatConfiguracionEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionEventos.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConfiguracionEventos) businessCatConfiguracionEventos.findById(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionEventos.findById: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/count", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionEventos count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionEventos.count. . .");  
 DominioCatConfiguracionEventos dom = new DominioCatConfiguracionEventos();  
 dom.setCount(businessCatConfiguracionEventos.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionEventos.count: " + ex);  
 DominioCatConfiguracionEventos dom = new DominioCatConfiguracionEventos();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/existsById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionEventos existsById(@RequestBody DominioCatConfiguracionEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionEventos.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatConfiguracionEventos) businessCatConfiguracionEventos.existsById(dom);  
 if (dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionEventos.existsById: " + ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatConfiguracionOpciones.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceCatConfiguracionOpciones;  
import mx.mbi.dgc.dominio.DominioCatConfiguracionOpciones;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.negocio.bitacora.DominioPlantaOpcionesDetalle;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatConfiguracionOpciones")  
public class RESTCatConfiguracionOpciones {  
 private static final Logger LOG = LogManager.getLogger(RESTCatConfiguracionOpciones.class);  
 @Autowired  
 private IfaceCatConfiguracionOpciones businessCatConfiguracionOpciones;  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAll", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioCatConfiguracionOpciones> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionOpciones.findAll. . .");  
 List<DominioCatConfiguracionOpciones> lista = businessCatConfiguracionOpciones.findAll();  
 LOG.info("invoke RESTCatConfiguracionOpciones.findAll size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionOpciones.findAll: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findByCatConfiguracionDetalle", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioCatConfiguracionOpciones> findByCatConfiguracionDetalle(@RequestBody  
 DominioPlantaOpcionesDetalle dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionOpciones.findByCatConfiguracionDetalle. . .");  
 List<DominioCatConfiguracionOpciones> lista =  
 businessCatConfiguracionOpciones.findByCatConfiguracionDetalle(dom);  
 LOG.info("invoke RESTCatConfiguracionOpciones.findAll size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionOpciones.findByCatConfiguracionDetalle: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAllPagination", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioRespuesta findAllPagination(@RequestBody DominioCatConfiguracionOpciones dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatConfiguracionOpciones.findAllPagination. . .");  
 response = businessCatConfiguracionOpciones.allPagination(dom);  
 LOG.info("invoke RESTCatConfiguracionOpciones.findAllPagination size lista return: " +  
 response.getData().size());  
 return response;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionOpciones.findAllPagination: " + ex);  
 response.setMensaje("Error in invoke RESTCatConfiguracionOpciones.findAllPagination: " + ex);  
 return response;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/save", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionOpciones save(@RequestBody DominioCatConfiguracionOpciones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionOpciones.save. . .");  
 LOG.info("dom.getConfiguracionOpcionesId(): " + dom.getConfiguracionDetalleId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConfiguracionOpciones) businessCatConfiguracionOpciones.save(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionOpciones.save: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/update", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionOpciones update(@RequestBody DominioCatConfiguracionOpciones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionOpciones.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConfiguracionOpciones) businessCatConfiguracionOpciones.update(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionOpciones.update: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/delete", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionOpciones delete(@RequestBody DominioCatConfiguracionOpciones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionOpciones.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConfiguracionOpciones) businessCatConfiguracionOpciones.delete(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionOpciones.delete: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionOpciones findById(@RequestBody DominioCatConfiguracionOpciones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionOpciones.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatConfiguracionOpciones) businessCatConfiguracionOpciones.findById(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionOpciones.findById: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/count", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionOpciones count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionOpciones.count. . .");  
 DominioCatConfiguracionOpciones dom = new DominioCatConfiguracionOpciones();  
 dom.setCount(businessCatConfiguracionOpciones.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionOpciones.count: " + ex);  
 DominioCatConfiguracionOpciones dom = new DominioCatConfiguracionOpciones();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/existsById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatConfiguracionOpciones existsById(@RequestBody DominioCatConfiguracionOpciones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatConfiguracionOpciones.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatConfiguracionOpciones) businessCatConfiguracionOpciones.existsById(dom);  
 if (dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatConfiguracionOpciones.existsById: " + ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatContratosAfectados.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceCatContratosAfectados;  
import mx.mbi.dgc.dominio.DominioCatContratosAfectados;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatContratosAfectados")  
public class RESTCatContratosAfectados {  
 private static final Logger LOG = LogManager.getLogger(RESTCatContratosAfectados.class);  
 @Autowired  
 private IfaceCatContratosAfectados businessCatContratosAfectados;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatContratosAfectados> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatContratosAfectados.findAll. . .");  
 List<DominioCatContratosAfectados> lista = businessCatContratosAfectados.findAll();  
 LOG.info("invoke RESTCatContratosAfectados.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatContratosAfectados.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatContratosAfectados dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatContratosAfectados.findAllPagination. . .");  
 response = businessCatContratosAfectados.allPagination(dom);  
 LOG.info("invoke RESTCatContratosAfectados.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatContratosAfectados.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatContratosAfectados.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatContratosAfectados save(@RequestBody DominioCatContratosAfectados dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatContratosAfectados.save. . .");  
 LOG.info("dom.getContratoAfectadoId(): "+dom.getContratoAfectadoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatContratosAfectados)businessCatContratosAfectados.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatContratosAfectados.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatContratosAfectados update(@RequestBody DominioCatContratosAfectados dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatContratosAfectados.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatContratosAfectados)businessCatContratosAfectados.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatContratosAfectados.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatContratosAfectados delete(@RequestBody DominioCatContratosAfectados dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatContratosAfectados.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatContratosAfectados)businessCatContratosAfectados.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatContratosAfectados.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatContratosAfectados findById(@RequestBody DominioCatContratosAfectados dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatContratosAfectados.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatContratosAfectados)businessCatContratosAfectados.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatContratosAfectados.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatContratosAfectados count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatContratosAfectados.count. . .");  
 DominioCatContratosAfectados dom = new DominioCatContratosAfectados();  
 dom.setCount(businessCatContratosAfectados.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatContratosAfectados.count: "+ex);  
 DominioCatContratosAfectados dom = new DominioCatContratosAfectados();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatContratosAfectados existsById(@RequestBody DominioCatContratosAfectados dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatContratosAfectados.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatContratosAfectados)businessCatContratosAfectados.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatContratosAfectados.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatCurvasCorreccion.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatCurvasCorreccion;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatCurvasCorreccion;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatCurvasCorreccion")  
public class RESTCatCurvasCorreccion {  
 private static final Logger LOG = LogManager.getLogger(RESTCatCurvasCorreccion.class);  
 @Autowired  
 private IfaceCatCurvasCorreccion businessCatCurvasCorreccion;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatCurvasCorreccion> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCurvasCorreccion.findAll. . .");  
 List<DominioCatCurvasCorreccion> lista = businessCatCurvasCorreccion.findAll();  
 LOG.info("invoke RESTCatCurvasCorreccion.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCurvasCorreccion.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatCurvasCorreccion dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatCurvasCorreccion.findAllPagination. . .");  
 response = businessCatCurvasCorreccion.allPagination(dom);  
 LOG.info("invoke RESTCatCurvasCorreccion.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCurvasCorreccion.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatCurvasCorreccion.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCurvasCorreccion save(@RequestBody DominioCatCurvasCorreccion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCurvasCorreccion.save. . .");  
 LOG.info("dom.getCurvaId(): "+dom.getCurvaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCurvasCorreccion)businessCatCurvasCorreccion.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCurvasCorreccion.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCurvasCorreccion update(@RequestBody DominioCatCurvasCorreccion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCurvasCorreccion.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCurvasCorreccion)businessCatCurvasCorreccion.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCurvasCorreccion.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCurvasCorreccion delete(@RequestBody DominioCatCurvasCorreccion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCurvasCorreccion.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCurvasCorreccion)businessCatCurvasCorreccion.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCurvasCorreccion.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCurvasCorreccion findById(@RequestBody DominioCatCurvasCorreccion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCurvasCorreccion.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatCurvasCorreccion)businessCatCurvasCorreccion.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCurvasCorreccion.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCurvasCorreccion count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatCurvasCorreccion.count. . .");  
 DominioCatCurvasCorreccion dom = new DominioCatCurvasCorreccion();  
 dom.setCount(businessCatCurvasCorreccion.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCurvasCorreccion.count: "+ex);  
 DominioCatCurvasCorreccion dom = new DominioCatCurvasCorreccion();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatCurvasCorreccion existsById(@RequestBody DominioCatCurvasCorreccion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatCurvasCorreccion.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatCurvasCorreccion)businessCatCurvasCorreccion.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatCurvasCorreccion.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatDatosFiscales.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatDatosFiscales;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatDatosFiscales;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatDatosFiscales")  
public class RESTCatDatosFiscales {  
 private static final Logger LOG = LogManager.getLogger(RESTCatDatosFiscales.class);  
 @Autowired  
 private IfaceCatDatosFiscales businessCatDatosFiscales;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatDatosFiscales> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatDatosFiscales.findAll. . .");  
 List<DominioCatDatosFiscales> lista = businessCatDatosFiscales.findAll();  
 LOG.info("invoke RESTCatDatosFiscales.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDatosFiscales.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatDatosFiscales dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatDatosFiscales.findAllPagination. . .");  
 response = businessCatDatosFiscales.allPagination(dom);  
 LOG.info("invoke RESTCatDatosFiscales.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDatosFiscales.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatDatosFiscales.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDatosFiscales save(@RequestBody DominioCatDatosFiscales dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDatosFiscales.save. . .");  
 LOG.info("dom.getDatosFiscalesId(): "+dom.getDatosFiscalesId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatDatosFiscales)businessCatDatosFiscales.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDatosFiscales.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDatosFiscales update(@RequestBody DominioCatDatosFiscales dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDatosFiscales.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatDatosFiscales)businessCatDatosFiscales.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDatosFiscales.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDatosFiscales delete(@RequestBody DominioCatDatosFiscales dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDatosFiscales.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatDatosFiscales)businessCatDatosFiscales.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDatosFiscales.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDatosFiscales findById(@RequestBody DominioCatDatosFiscales dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDatosFiscales.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatDatosFiscales)businessCatDatosFiscales.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDatosFiscales.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDatosFiscales count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatDatosFiscales.count. . .");  
 DominioCatDatosFiscales dom = new DominioCatDatosFiscales();  
 dom.setCount(businessCatDatosFiscales.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDatosFiscales.count: "+ex);  
 DominioCatDatosFiscales dom = new DominioCatDatosFiscales();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDatosFiscales existsById(@RequestBody DominioCatDatosFiscales dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDatosFiscales.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatDatosFiscales)businessCatDatosFiscales.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDatosFiscales.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatDetalleOpciones.java

package mx.mbi.dgc.microservices.controller;  
  
import java.math.BigDecimal;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceCatDetalleOpciones;  
import mx.mbi.dgc.dominio.DominioCatDetalleOpciones;  
import mx.mbi.dgc.dominio.DominioCatDetalles;  
import mx.mbi.dgc.dominio.DominioCatOpcionPlanta;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
  
import mx.mbi.dgc.business.IfaceCatOpcionPlanta;  
  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatDetalleOpciones")  
public class RESTCatDetalleOpciones {  
 private static final Logger LOG = LogManager.getLogger(RESTCatDetalleOpciones.class);  
 @Autowired  
 private IfaceCatDetalleOpciones businessCatDetalleOpciones;  
 @Autowired  
 private IfaceCatOpcionPlanta businessCatOpcionPlanta;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatDetalleOpciones> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalleOpciones.findAll. . .");  
 List<DominioCatDetalleOpciones> lista = businessCatDetalleOpciones.findAll();  
 LOG.info("invoke RESTCatDetalleOpciones.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalleOpciones.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }   
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findByCatDetalle",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatDetalleOpciones> findByDetalleId(@RequestBody DominioCatDetalles dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalleOpciones.findByDetalleId. . .");  
 List<DominioCatDetalleOpciones> lista = businessCatDetalleOpciones.findByDetalleId(dom);  
 LOG.info("invoke RESTCatDetalleOpciones.findByDetalleId size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalleOpciones.findByDetalleId: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findByDetallePlantaId",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatDetalleOpciones> findByDetallePlantaId(@RequestBody DominioCatOpcionPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalleOpciones.findByDetallePlantaId. . .");  
 List<DominioCatDetalleOpciones> lista = businessCatDetalleOpciones.findByDetallePlantaId(dom);  
 LOG.info("invoke RESTCatDetalleOpciones.findByDetalleId size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalleOpciones.findByDetalleId: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findCheck",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatDetalleOpciones> findCheck (@RequestBody DominioCatOpcionPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalleOpciones.findByDetallePlantaId. . .");  
 List<DominioCatDetalleOpciones> lista = businessCatDetalleOpciones.findCheck(dom);  
 LOG.info("invoke RESTCatDetalleOpciones.findByDetalleId size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalleOpciones.findByDetalleId: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findByDetalleId",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatDetalleOpciones> findByDetalle(@RequestBody DominioCatDetalles dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalleOpciones.findByDetalleId. . .");  
 List<DominioCatDetalleOpciones> lista = businessCatDetalleOpciones.findByDetalle(dom);  
 LOG.info("invoke RESTCatDetalleOpciones.findByDetalleId size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalleOpciones.findByDetalleId: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatDetalleOpciones dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatDetalleOpciones.findAllPagination. . .");  
 response = businessCatDetalleOpciones.allPagination(dom);  
 LOG.info("invoke RESTCatDetalleOpciones.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalleOpciones.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatDetalleOpciones.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDetalleOpciones save(@RequestBody DominioCatDetalleOpciones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalleOpciones.save. . .");  
 LOG.info("dom.getDetalleOpcionesId(): "+dom.getDetalleOpcionesId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 long id = businessCatDetalleOpciones.count();  
 id = id + 2;  
 dom.setDetalleOpcionesId(BigDecimal.valueOf(id));  
 return (DominioCatDetalleOpciones)businessCatDetalleOpciones.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalleOpciones.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDetalleOpciones update(@RequestBody DominioCatDetalleOpciones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalleOpciones.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatDetalleOpciones)businessCatDetalleOpciones.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalleOpciones.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDetalleOpciones delete(@RequestBody DominioCatOpcionPlanta dom2) throws Exception {  
 DominioCatDetalleOpciones dom = new DominioCatDetalleOpciones();  
 try {  
 dom.setDetalleOpcionesId(dom2.getDetalleOpcionesId().getDetalleOpcionesId());  
 LOG.info("invoke RESTCatDetalleOpciones.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom2 = businessCatOpcionPlanta.findByOpcionPlanta(dom2);  
 businessCatOpcionPlanta.delete(dom2);  
 return (DominioCatDetalleOpciones)businessCatDetalleOpciones.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalleOpciones.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDetalleOpciones findById(@RequestBody DominioCatDetalleOpciones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalleOpciones.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatDetalleOpciones)businessCatDetalleOpciones.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalleOpciones.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDetalleOpciones count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalleOpciones.count. . .");  
 DominioCatDetalleOpciones dom = new DominioCatDetalleOpciones();  
 dom.setCount(businessCatDetalleOpciones.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalleOpciones.count: "+ex);  
 DominioCatDetalleOpciones dom = new DominioCatDetalleOpciones();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDetalleOpciones existsById(@RequestBody DominioCatDetalleOpciones dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalleOpciones.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatDetalleOpciones)businessCatDetalleOpciones.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalleOpciones.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatDetalles.java

package mx.mbi.dgc.microservices.controller;  
  
import java.math.BigDecimal;  
  
import java.util.ArrayList;  
import java.util.List;  
  
  
import mx.mbi.dgc.business.IfaceCatDetalles;  
import mx.mbi.dgc.dominio.DominioCatDetalles;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatDetalles")  
public class RESTCatDetalles {  
 private static final Logger LOG = LogManager.getLogger(RESTCatDetalles.class);  
 @Autowired  
 private IfaceCatDetalles businessCatDetalles;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatDetalles> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalles.findAll. . .");  
 List<DominioCatDetalles> lista = businessCatDetalles.findAll();  
 LOG.info("invoke RESTCatDetalles.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalles.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }   
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatDetalles dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatDetalles.findAllPagination. . .");  
 response = businessCatDetalles.allPagination(dom);  
 LOG.info("invoke RESTCatDetalles.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalles.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatDetalles.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDetalles save(@RequestBody DominioCatDetalles dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalles.save. . .");  
 LOG.info("dom.getDetalleId(): "+dom.getDetalleId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 long id = businessCatDetalles.count();  
 id = id + 1;  
 dom.setDetalleId(BigDecimal.valueOf(id));  
 return (DominioCatDetalles)businessCatDetalles.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalles.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDetalles update(@RequestBody DominioCatDetalles dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalles.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatDetalles)businessCatDetalles.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalles.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDetalles delete(@RequestBody DominioCatDetalles dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalles.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatDetalles)businessCatDetalles.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalles.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDetalles findById(@RequestBody DominioCatDetalles dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalles.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatDetalles)businessCatDetalles.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalles.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDetalles count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalles.count. . .");  
 DominioCatDetalles dom = new DominioCatDetalles();  
 dom.setCount(businessCatDetalles.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalles.count: "+ex);  
 DominioCatDetalles dom = new DominioCatDetalles();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatDetalles existsById(@RequestBody DominioCatDetalles dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatDetalles.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatDetalles)businessCatDetalles.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatDetalles.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatEstados.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatEstados;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatEstados;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatEstados")  
public class RESTCatEstados {  
 private static final Logger LOG = LogManager.getLogger(RESTCatEstados.class);  
 @Autowired  
 private IfaceCatEstados businessCatEstados;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatEstados> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstados.findAll. . .");  
 List<DominioCatEstados> lista = businessCatEstados.findAll();  
 LOG.info("invoke RESTCatEstados.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstados.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatEstados dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatEstados.findAllPagination. . .");  
 response = businessCatEstados.allPagination(dom);  
 LOG.info("invoke RESTCatEstados.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstados.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatEstados.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstados save(@RequestBody DominioCatEstados dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstados.save. . .");  
 LOG.info("dom.getEstadoId(): "+dom.getEstadoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstados)businessCatEstados.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstados.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstados update(@RequestBody DominioCatEstados dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstados.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstados)businessCatEstados.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstados.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstados delete(@RequestBody DominioCatEstados dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstados.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstados)businessCatEstados.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstados.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstados findById(@RequestBody DominioCatEstados dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstados.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstados)businessCatEstados.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstados.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstados count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstados.count. . .");  
 DominioCatEstados dom = new DominioCatEstados();  
 dom.setCount(businessCatEstados.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstados.count: "+ex);  
 DominioCatEstados dom = new DominioCatEstados();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstados existsById(@RequestBody DominioCatEstados dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstados.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatEstados)businessCatEstados.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstados.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatEstatusBitacora.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatEstatusBitacora;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatEstatusBitacora;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatEstatusBitacora")  
public class RESTCatEstatusBitacora {  
 private static final Logger LOG = LogManager.getLogger(RESTCatEstatusBitacora.class);  
 @Autowired  
 private IfaceCatEstatusBitacora businessCatEstatusBitacora;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatEstatusBitacora> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusBitacora.findAll. . .");  
 List<DominioCatEstatusBitacora> lista = businessCatEstatusBitacora.findAll();  
 LOG.info("invoke RESTCatEstatusBitacora.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusBitacora.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatEstatusBitacora dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatEstatusBitacora.findAllPagination. . .");  
 response = businessCatEstatusBitacora.allPagination(dom);  
 LOG.info("invoke RESTCatEstatusBitacora.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusBitacora.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatEstatusBitacora.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusBitacora save(@RequestBody DominioCatEstatusBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusBitacora.save. . .");  
 LOG.info("dom.getEstatusBitacoraId(): "+dom.getEstatusBitacoraId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusBitacora)businessCatEstatusBitacora.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusBitacora.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusBitacora update(@RequestBody DominioCatEstatusBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusBitacora.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusBitacora)businessCatEstatusBitacora.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusBitacora.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusBitacora delete(@RequestBody DominioCatEstatusBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusBitacora.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusBitacora)businessCatEstatusBitacora.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusBitacora.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusBitacora findById(@RequestBody DominioCatEstatusBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusBitacora.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusBitacora)businessCatEstatusBitacora.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusBitacora.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusBitacora count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusBitacora.count. . .");  
 DominioCatEstatusBitacora dom = new DominioCatEstatusBitacora();  
 dom.setCount(businessCatEstatusBitacora.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusBitacora.count: "+ex);  
 DominioCatEstatusBitacora dom = new DominioCatEstatusBitacora();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusBitacora existsById(@RequestBody DominioCatEstatusBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusBitacora.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatEstatusBitacora)businessCatEstatusBitacora.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusBitacora.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatEstatusFactura.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatEstatusFactura;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatEstatusFactura;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatEstatusFactura")  
public class RESTCatEstatusFactura {  
 private static final Logger LOG = LogManager.getLogger(RESTCatEstatusFactura.class);  
 @Autowired  
 private IfaceCatEstatusFactura businessCatEstatusFactura;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatEstatusFactura> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusFactura.findAll. . .");  
 List<DominioCatEstatusFactura> lista = businessCatEstatusFactura.findAll();  
 LOG.info("invoke RESTCatEstatusFactura.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusFactura.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatEstatusFactura dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatEstatusFactura.findAllPagination. . .");  
 response = businessCatEstatusFactura.allPagination(dom);  
 LOG.info("invoke RESTCatEstatusFactura.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusFactura.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatEstatusFactura.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusFactura save(@RequestBody DominioCatEstatusFactura dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusFactura.save. . .");  
 LOG.info("dom.getEstatusFacturaId(): "+dom.getEstatusFacturaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusFactura)businessCatEstatusFactura.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusFactura.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusFactura update(@RequestBody DominioCatEstatusFactura dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusFactura.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusFactura)businessCatEstatusFactura.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusFactura.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusFactura delete(@RequestBody DominioCatEstatusFactura dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusFactura.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusFactura)businessCatEstatusFactura.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusFactura.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusFactura findById(@RequestBody DominioCatEstatusFactura dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusFactura.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusFactura)businessCatEstatusFactura.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusFactura.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusFactura count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusFactura.count. . .");  
 DominioCatEstatusFactura dom = new DominioCatEstatusFactura();  
 dom.setCount(businessCatEstatusFactura.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusFactura.count: "+ex);  
 DominioCatEstatusFactura dom = new DominioCatEstatusFactura();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusFactura existsById(@RequestBody DominioCatEstatusFactura dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusFactura.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatEstatusFactura)businessCatEstatusFactura.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusFactura.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatEstatusPagos.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatEstatusPagos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatEstatusPagos;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatEstatusPagos")  
public class RESTCatEstatusPagos {  
 private static final Logger LOG = LogManager.getLogger(RESTCatEstatusPagos.class);  
 @Autowired  
 private IfaceCatEstatusPagos businessCatEstatusPagos;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatEstatusPagos> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusPagos.findAll. . .");  
 List<DominioCatEstatusPagos> lista = businessCatEstatusPagos.findAll();  
 LOG.info("invoke RESTCatEstatusPagos.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPagos.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatEstatusPagos dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatEstatusPagos.findAllPagination. . .");  
 response = businessCatEstatusPagos.allPagination(dom);  
 LOG.info("invoke RESTCatEstatusPagos.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPagos.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatEstatusPagos.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusPagos save(@RequestBody DominioCatEstatusPagos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusPagos.save. . .");  
 LOG.info("dom.getEstatusPagoId(): "+dom.getEstatusPagoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusPagos)businessCatEstatusPagos.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPagos.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusPagos update(@RequestBody DominioCatEstatusPagos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusPagos.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusPagos)businessCatEstatusPagos.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPagos.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusPagos delete(@RequestBody DominioCatEstatusPagos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusPagos.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusPagos)businessCatEstatusPagos.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPagos.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusPagos findById(@RequestBody DominioCatEstatusPagos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusPagos.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusPagos)businessCatEstatusPagos.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPagos.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusPagos count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusPagos.count. . .");  
 DominioCatEstatusPagos dom = new DominioCatEstatusPagos();  
 dom.setCount(businessCatEstatusPagos.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPagos.count: "+ex);  
 DominioCatEstatusPagos dom = new DominioCatEstatusPagos();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusPagos existsById(@RequestBody DominioCatEstatusPagos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusPagos.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatEstatusPagos)businessCatEstatusPagos.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPagos.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatEstatusPerfil.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatEstatusPerfil;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatEstatusPerfil;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatEstatusPerfil")  
public class RESTCatEstatusPerfil {  
 private static final Logger LOG = LogManager.getLogger(RESTCatEstatusPerfil.class);  
 @Autowired  
 private IfaceCatEstatusPerfil businessCatEstatusPerfil;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatEstatusPerfil> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusPerfil.findAll. . .");  
 List<DominioCatEstatusPerfil> lista = businessCatEstatusPerfil.findAll();  
 LOG.info("invoke RESTCatEstatusPerfil.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPerfil.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatEstatusPerfil dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatEstatusPerfil.findAllPagination. . .");  
 response = businessCatEstatusPerfil.allPagination(dom);  
 LOG.info("invoke RESTCatEstatusPerfil.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPerfil.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatEstatusPerfil.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusPerfil save(@RequestBody DominioCatEstatusPerfil dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusPerfil.save. . .");  
 LOG.info("dom.getEstatusPerfilId(): "+dom.getEstatusPerfilId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusPerfil)businessCatEstatusPerfil.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPerfil.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusPerfil update(@RequestBody DominioCatEstatusPerfil dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusPerfil.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusPerfil)businessCatEstatusPerfil.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPerfil.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusPerfil delete(@RequestBody DominioCatEstatusPerfil dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusPerfil.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusPerfil)businessCatEstatusPerfil.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPerfil.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusPerfil findById(@RequestBody DominioCatEstatusPerfil dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusPerfil.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusPerfil)businessCatEstatusPerfil.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPerfil.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusPerfil count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusPerfil.count. . .");  
 DominioCatEstatusPerfil dom = new DominioCatEstatusPerfil();  
 dom.setCount(businessCatEstatusPerfil.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPerfil.count: "+ex);  
 DominioCatEstatusPerfil dom = new DominioCatEstatusPerfil();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusPerfil existsById(@RequestBody DominioCatEstatusPerfil dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusPerfil.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatEstatusPerfil)businessCatEstatusPerfil.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusPerfil.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatEstatusVariable.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatEstatusVariable;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatEstatusVariable;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatEstatusVariable")  
public class RESTCatEstatusVariable {  
 private static final Logger LOG = LogManager.getLogger(RESTCatEstatusVariable.class);  
 @Autowired  
 private IfaceCatEstatusVariable businessCatEstatusVariable;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatEstatusVariable> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusVariable.findAll. . .");  
 List<DominioCatEstatusVariable> lista = businessCatEstatusVariable.findAll();  
 LOG.info("invoke RESTCatEstatusVariable.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusVariable.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatEstatusVariable dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatEstatusVariable.findAllPagination. . .");  
 response = businessCatEstatusVariable.allPagination(dom);  
 LOG.info("invoke RESTCatEstatusVariable.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusVariable.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatEstatusVariable.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusVariable save(@RequestBody DominioCatEstatusVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusVariable.save. . .");  
 LOG.info("dom.getEstatusVariableId(): "+dom.getEstatusVariableId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusVariable)businessCatEstatusVariable.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusVariable.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusVariable update(@RequestBody DominioCatEstatusVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusVariable.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusVariable)businessCatEstatusVariable.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusVariable.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusVariable delete(@RequestBody DominioCatEstatusVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusVariable.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusVariable)businessCatEstatusVariable.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusVariable.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusVariable findById(@RequestBody DominioCatEstatusVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusVariable.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEstatusVariable)businessCatEstatusVariable.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusVariable.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusVariable count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusVariable.count. . .");  
 DominioCatEstatusVariable dom = new DominioCatEstatusVariable();  
 dom.setCount(businessCatEstatusVariable.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusVariable.count: "+ex);  
 DominioCatEstatusVariable dom = new DominioCatEstatusVariable();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEstatusVariable existsById(@RequestBody DominioCatEstatusVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEstatusVariable.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatEstatusVariable)businessCatEstatusVariable.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEstatusVariable.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatEtapas.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatEtapas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatEtapas;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatEtapas")  
public class RESTCatEtapas {  
 private static final Logger LOG = LogManager.getLogger(RESTCatEtapas.class);  
 @Autowired  
 private IfaceCatEtapas businessCatEtapas;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatEtapas> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEtapas.findAll. . .");  
 List<DominioCatEtapas> lista = businessCatEtapas.findAll();  
 LOG.info("invoke RESTCatEtapas.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEtapas.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatEtapas dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatEtapas.findAllPagination. . .");  
 response = businessCatEtapas.allPagination(dom);  
 LOG.info("invoke RESTCatEtapas.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEtapas.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatEtapas.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEtapas save(@RequestBody DominioCatEtapas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEtapas.save. . .");  
 LOG.info("dom.getEtapaId(): "+dom.getEtapaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEtapas)businessCatEtapas.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEtapas.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEtapas update(@RequestBody DominioCatEtapas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEtapas.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEtapas)businessCatEtapas.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEtapas.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEtapas delete(@RequestBody DominioCatEtapas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEtapas.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEtapas)businessCatEtapas.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEtapas.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEtapas findById(@RequestBody DominioCatEtapas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEtapas.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEtapas)businessCatEtapas.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEtapas.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEtapas count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEtapas.count. . .");  
 DominioCatEtapas dom = new DominioCatEtapas();  
 dom.setCount(businessCatEtapas.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEtapas.count: "+ex);  
 DominioCatEtapas dom = new DominioCatEtapas();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEtapas existsById(@RequestBody DominioCatEtapas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEtapas.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatEtapas)businessCatEtapas.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEtapas.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatEventos.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceCatEventos;  
import mx.mbi.dgc.dominio.DominioCatEventos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatEventos")  
public class RESTCatEventos {  
 private static final Logger LOG = LogManager.getLogger(RESTCatEventos.class);  
 @Autowired  
 private IfaceCatEventos businessCatEventos;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatEventos> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEventos.findAll. . .");  
 List<DominioCatEventos> lista = businessCatEventos.findAll();  
 LOG.info("invoke RESTCatEventos.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEventos.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatEventos dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatEventos.findAllPagination. . .");  
 response = businessCatEventos.allPagination(dom);  
 LOG.info("invoke RESTCatEventos.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEventos.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatEventos.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEventos save(@RequestBody DominioCatEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEventos.save. . .");  
 LOG.info("dom.getEventoId(): "+dom.getEventoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEventos)businessCatEventos.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEventos.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEventos update(@RequestBody DominioCatEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEventos.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEventos)businessCatEventos.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEventos.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEventos delete(@RequestBody DominioCatEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEventos.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEventos)businessCatEventos.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEventos.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEventos findById(@RequestBody DominioCatEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEventos.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatEventos)businessCatEventos.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEventos.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEventos count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatEventos.count. . .");  
 DominioCatEventos dom = new DominioCatEventos();  
 dom.setCount(businessCatEventos.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEventos.count: "+ex);  
 DominioCatEventos dom = new DominioCatEventos();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatEventos existsById(@RequestBody DominioCatEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatEventos.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatEventos)businessCatEventos.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatEventos.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatFactorTipo.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatFactorTipo;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatFactorTipo;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatFactorTipo")  
public class RESTCatFactorTipo {  
 private static final Logger LOG = LogManager.getLogger(RESTCatFactorTipo.class);  
 @Autowired  
 private IfaceCatFactorTipo businessCatFactorTipo;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatFactorTipo> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatFactorTipo.findAll. . .");  
 List<DominioCatFactorTipo> lista = businessCatFactorTipo.findAll();  
 LOG.info("invoke RESTCatFactorTipo.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFactorTipo.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatFactorTipo dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatFactorTipo.findAllPagination. . .");  
 response = businessCatFactorTipo.allPagination(dom);  
 LOG.info("invoke RESTCatFactorTipo.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFactorTipo.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatFactorTipo.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFactorTipo save(@RequestBody DominioCatFactorTipo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFactorTipo.save. . .");  
 LOG.info("dom.getFactorTipoId(): "+dom.getFactorTipoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFactorTipo)businessCatFactorTipo.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFactorTipo.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFactorTipo update(@RequestBody DominioCatFactorTipo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFactorTipo.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFactorTipo)businessCatFactorTipo.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFactorTipo.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFactorTipo delete(@RequestBody DominioCatFactorTipo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFactorTipo.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFactorTipo)businessCatFactorTipo.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFactorTipo.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFactorTipo findById(@RequestBody DominioCatFactorTipo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFactorTipo.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFactorTipo)businessCatFactorTipo.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFactorTipo.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFactorTipo count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatFactorTipo.count. . .");  
 DominioCatFactorTipo dom = new DominioCatFactorTipo();  
 dom.setCount(businessCatFactorTipo.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFactorTipo.count: "+ex);  
 DominioCatFactorTipo dom = new DominioCatFactorTipo();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFactorTipo existsById(@RequestBody DominioCatFactorTipo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFactorTipo.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatFactorTipo)businessCatFactorTipo.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFactorTipo.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatFacturaProductos.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatFacturaProductos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatFacturaProductos;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatFacturaProductos")  
public class RESTCatFacturaProductos {  
 private static final Logger LOG = LogManager.getLogger(RESTCatFacturaProductos.class);  
 @Autowired  
 private IfaceCatFacturaProductos businessCatFacturaProductos;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatFacturaProductos> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatFacturaProductos.findAll. . .");  
 List<DominioCatFacturaProductos> lista = businessCatFacturaProductos.findAll();  
 LOG.info("invoke RESTCatFacturaProductos.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFacturaProductos.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatFacturaProductos dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatFacturaProductos.findAllPagination. . .");  
 response = businessCatFacturaProductos.allPagination(dom);  
 LOG.info("invoke RESTCatFacturaProductos.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFacturaProductos.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatFacturaProductos.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFacturaProductos save(@RequestBody DominioCatFacturaProductos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFacturaProductos.save. . .");  
 LOG.info("dom.getFacturaProductoId(): "+dom.getFacturaProductoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFacturaProductos)businessCatFacturaProductos.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFacturaProductos.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFacturaProductos update(@RequestBody DominioCatFacturaProductos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFacturaProductos.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFacturaProductos)businessCatFacturaProductos.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFacturaProductos.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFacturaProductos delete(@RequestBody DominioCatFacturaProductos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFacturaProductos.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFacturaProductos)businessCatFacturaProductos.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFacturaProductos.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFacturaProductos findById(@RequestBody DominioCatFacturaProductos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFacturaProductos.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFacturaProductos)businessCatFacturaProductos.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFacturaProductos.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFacturaProductos count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatFacturaProductos.count. . .");  
 DominioCatFacturaProductos dom = new DominioCatFacturaProductos();  
 dom.setCount(businessCatFacturaProductos.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFacturaProductos.count: "+ex);  
 DominioCatFacturaProductos dom = new DominioCatFacturaProductos();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFacturaProductos existsById(@RequestBody DominioCatFacturaProductos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFacturaProductos.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatFacturaProductos)businessCatFacturaProductos.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFacturaProductos.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatFirmaCincominutal.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceCatFirmaCincominutal;  
import mx.mbi.dgc.dominio.DominioCatFirmaCincominutal;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatFirmaCincominutal")  
public class RESTCatFirmaCincominutal {  
 private static final Logger LOG = LogManager.getLogger(RESTCatFirmaCincominutal.class);  
 @Autowired  
 private IfaceCatFirmaCincominutal businessCatFirmaCincominutal;  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAll", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioCatFirmaCincominutal> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatFirmaCincominutal.findAll. . .");  
 List<DominioCatFirmaCincominutal> lista = businessCatFirmaCincominutal.findAll();  
 LOG.info("invoke RESTCatFirmaCincominutal.findAll size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatFirmaCincominutal.findAll: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAllLimit", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioCatFirmaCincominutal> findAllLimit() throws Exception {  
 try {  
 LOG.info("invoke RESTCatFirmaCincominutal.findAllLimit. . .");  
 List<DominioCatFirmaCincominutal> lista = businessCatFirmaCincominutal.findAllLimit();  
 LOG.info("invoke RESTCatFirmaCincominutal.findAllLimit size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 LOG.error("Error in invoke RESTCatFirmaCincominutal.findAllLimit: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAllPagination", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioRespuesta findAllPagination(@RequestBody DominioCatFirmaCincominutal dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatFirmaCincominutal.findAllPagination. . .");  
 response = businessCatFirmaCincominutal.allPagination(dom);  
 LOG.info("invoke RESTCatFirmaCincominutal.findAllPagination size lista return: " +  
 response.getData().size());  
 return response;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatFirmaCincominutal.findAllPagination: " + ex);  
 response.setMensaje("Error in invoke RESTCatFirmaCincominutal.findAllPagination: " + ex);  
 return response;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/save", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatFirmaCincominutal save(@RequestBody DominioCatFirmaCincominutal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFirmaCincominutal.save. . .");  
 LOG.info("dom.getFirmaId(): " + dom.getFirmaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFirmaCincominutal) businessCatFirmaCincominutal.save(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatFirmaCincominutal.save: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/update", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatFirmaCincominutal update(@RequestBody DominioCatFirmaCincominutal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFirmaCincominutal.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFirmaCincominutal) businessCatFirmaCincominutal.update(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatFirmaCincominutal.update: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/delete", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatFirmaCincominutal delete(@RequestBody DominioCatFirmaCincominutal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFirmaCincominutal.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFirmaCincominutal) businessCatFirmaCincominutal.delete(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatFirmaCincominutal.delete: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatFirmaCincominutal findById(@RequestBody DominioCatFirmaCincominutal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFirmaCincominutal.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFirmaCincominutal) businessCatFirmaCincominutal.findById(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatFirmaCincominutal.findById: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/count", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatFirmaCincominutal count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatFirmaCincominutal.count. . .");  
 DominioCatFirmaCincominutal dom = new DominioCatFirmaCincominutal();  
 dom.setCount(businessCatFirmaCincominutal.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatFirmaCincominutal.count: " + ex);  
 DominioCatFirmaCincominutal dom = new DominioCatFirmaCincominutal();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/existsById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatFirmaCincominutal existsById(@RequestBody DominioCatFirmaCincominutal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFirmaCincominutal.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatFirmaCincominutal) businessCatFirmaCincominutal.existsById(dom);  
 if (dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatFirmaCincominutal.existsById: " + ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatFormasPago.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatFormasPago;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatFormasPago;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatFormasPago")  
public class RESTCatFormasPago {  
 private static final Logger LOG = LogManager.getLogger(RESTCatFormasPago.class);  
 @Autowired  
 private IfaceCatFormasPago businessCatFormasPago;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatFormasPago> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormasPago.findAll. . .");  
 List<DominioCatFormasPago> lista = businessCatFormasPago.findAll();  
 LOG.info("invoke RESTCatFormasPago.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormasPago.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatFormasPago dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatFormasPago.findAllPagination. . .");  
 response = businessCatFormasPago.allPagination(dom);  
 LOG.info("invoke RESTCatFormasPago.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormasPago.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatFormasPago.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormasPago save(@RequestBody DominioCatFormasPago dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormasPago.save. . .");  
 LOG.info("dom.getFormaPagoId(): "+dom.getFormaPagoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFormasPago)businessCatFormasPago.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormasPago.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormasPago update(@RequestBody DominioCatFormasPago dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormasPago.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFormasPago)businessCatFormasPago.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormasPago.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormasPago delete(@RequestBody DominioCatFormasPago dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormasPago.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFormasPago)businessCatFormasPago.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormasPago.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormasPago findById(@RequestBody DominioCatFormasPago dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormasPago.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFormasPago)businessCatFormasPago.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormasPago.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormasPago count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormasPago.count. . .");  
 DominioCatFormasPago dom = new DominioCatFormasPago();  
 dom.setCount(businessCatFormasPago.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormasPago.count: "+ex);  
 DominioCatFormasPago dom = new DominioCatFormasPago();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormasPago existsById(@RequestBody DominioCatFormasPago dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormasPago.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatFormasPago)businessCatFormasPago.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormasPago.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatFormulaComponentes.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatFormulaComponentes;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatFormulaComponentes;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatFormulaComponentes")  
public class RESTCatFormulaComponentes {  
 private static final Logger LOG = LogManager.getLogger(RESTCatFormulaComponentes.class);  
 @Autowired  
 private IfaceCatFormulaComponentes businessCatFormulaComponentes;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatFormulaComponentes> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormulaComponentes.findAll. . .");  
 List<DominioCatFormulaComponentes> lista = businessCatFormulaComponentes.findAll();  
 LOG.info("invoke RESTCatFormulaComponentes.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentes.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatFormulaComponentes dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatFormulaComponentes.findAllPagination. . .");  
 response = businessCatFormulaComponentes.allPagination(dom);  
 LOG.info("invoke RESTCatFormulaComponentes.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentes.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatFormulaComponentes.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormulaComponentes save(@RequestBody DominioCatFormulaComponentes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormulaComponentes.save. . .");  
 LOG.info("dom.getCatFormulaComponentesId(): "+dom.getCatFormulaComponentesId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFormulaComponentes)businessCatFormulaComponentes.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentes.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormulaComponentes update(@RequestBody DominioCatFormulaComponentes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormulaComponentes.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFormulaComponentes)businessCatFormulaComponentes.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentes.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormulaComponentes delete(@RequestBody DominioCatFormulaComponentes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormulaComponentes.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFormulaComponentes)businessCatFormulaComponentes.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentes.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormulaComponentes findById(@RequestBody DominioCatFormulaComponentes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormulaComponentes.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFormulaComponentes)businessCatFormulaComponentes.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentes.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormulaComponentes count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormulaComponentes.count. . .");  
 DominioCatFormulaComponentes dom = new DominioCatFormulaComponentes();  
 dom.setCount(businessCatFormulaComponentes.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentes.count: "+ex);  
 DominioCatFormulaComponentes dom = new DominioCatFormulaComponentes();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormulaComponentes existsById(@RequestBody DominioCatFormulaComponentes dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormulaComponentes.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatFormulaComponentes)businessCatFormulaComponentes.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentes.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatFormulaComponentesTxEvento.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatFormulaComponentesTxEvento;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatFormulaComponentesTxEvento;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatFormulaComponentesTxEvento")  
public class RESTCatFormulaComponentesTxEvento {  
 private static final Logger LOG = LogManager.getLogger(RESTCatFormulaComponentesTxEvento.class);  
 @Autowired  
 private IfaceCatFormulaComponentesTxEvento businessCatFormulaComponentesTxEvento;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatFormulaComponentesTxEvento> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormulaComponentesTxEvento.findAll. . .");  
 List<DominioCatFormulaComponentesTxEvento> lista = businessCatFormulaComponentesTxEvento.findAll();  
 LOG.info("invoke RESTCatFormulaComponentesTxEvento.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentesTxEvento.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatFormulaComponentesTxEvento dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatFormulaComponentesTxEvento.findAllPagination. . .");  
 response = businessCatFormulaComponentesTxEvento.allPagination(dom);  
 LOG.info("invoke RESTCatFormulaComponentesTxEvento.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentesTxEvento.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatFormulaComponentesTxEvento.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormulaComponentesTxEvento save(@RequestBody DominioCatFormulaComponentesTxEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormulaComponentesTxEvento.save. . .");  
 LOG.info("dom.getCatFormulaComponentesTxEventoId(): "+dom.getCatFormulaComponentesTxEventoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFormulaComponentesTxEvento)businessCatFormulaComponentesTxEvento.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentesTxEvento.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormulaComponentesTxEvento update(@RequestBody DominioCatFormulaComponentesTxEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormulaComponentesTxEvento.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFormulaComponentesTxEvento)businessCatFormulaComponentesTxEvento.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentesTxEvento.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormulaComponentesTxEvento delete(@RequestBody DominioCatFormulaComponentesTxEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormulaComponentesTxEvento.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFormulaComponentesTxEvento)businessCatFormulaComponentesTxEvento.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentesTxEvento.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormulaComponentesTxEvento findById(@RequestBody DominioCatFormulaComponentesTxEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormulaComponentesTxEvento.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatFormulaComponentesTxEvento)businessCatFormulaComponentesTxEvento.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentesTxEvento.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormulaComponentesTxEvento count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormulaComponentesTxEvento.count. . .");  
 DominioCatFormulaComponentesTxEvento dom = new DominioCatFormulaComponentesTxEvento();  
 dom.setCount(businessCatFormulaComponentesTxEvento.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentesTxEvento.count: "+ex);  
 DominioCatFormulaComponentesTxEvento dom = new DominioCatFormulaComponentesTxEvento();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatFormulaComponentesTxEvento existsById(@RequestBody DominioCatFormulaComponentesTxEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatFormulaComponentesTxEvento.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatFormulaComponentesTxEvento)businessCatFormulaComponentesTxEvento.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatFormulaComponentesTxEvento.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatImpuestos.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatImpuestos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatImpuestos;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatImpuestos")  
public class RESTCatImpuestos {  
 private static final Logger LOG = LogManager.getLogger(RESTCatImpuestos.class);  
 @Autowired  
 private IfaceCatImpuestos businessCatImpuestos;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatImpuestos> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatImpuestos.findAll. . .");  
 List<DominioCatImpuestos> lista = businessCatImpuestos.findAll();  
 LOG.info("invoke RESTCatImpuestos.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatImpuestos.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatImpuestos dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatImpuestos.findAllPagination. . .");  
 response = businessCatImpuestos.allPagination(dom);  
 LOG.info("invoke RESTCatImpuestos.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatImpuestos.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatImpuestos.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatImpuestos save(@RequestBody DominioCatImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatImpuestos.save. . .");  
 LOG.info("dom.getImpuestoId(): "+dom.getImpuestoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatImpuestos)businessCatImpuestos.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatImpuestos.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatImpuestos update(@RequestBody DominioCatImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatImpuestos.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatImpuestos)businessCatImpuestos.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatImpuestos.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatImpuestos delete(@RequestBody DominioCatImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatImpuestos.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatImpuestos)businessCatImpuestos.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatImpuestos.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatImpuestos findById(@RequestBody DominioCatImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatImpuestos.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatImpuestos)businessCatImpuestos.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatImpuestos.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatImpuestos count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatImpuestos.count. . .");  
 DominioCatImpuestos dom = new DominioCatImpuestos();  
 dom.setCount(businessCatImpuestos.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatImpuestos.count: "+ex);  
 DominioCatImpuestos dom = new DominioCatImpuestos();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatImpuestos existsById(@RequestBody DominioCatImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatImpuestos.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatImpuestos)businessCatImpuestos.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatImpuestos.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatLimitesDominio.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatLimitesDominio;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatLimitesDominio;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatLimitesDominio")  
public class RESTCatLimitesDominio {  
 private static final Logger LOG = LogManager.getLogger(RESTCatLimitesDominio.class);  
 @Autowired  
 private IfaceCatLimitesDominio businessCatLimitesDominio;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatLimitesDominio> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatLimitesDominio.findAll. . .");  
 List<DominioCatLimitesDominio> lista = businessCatLimitesDominio.findAll();  
 LOG.info("invoke RESTCatLimitesDominio.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatLimitesDominio.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatLimitesDominio dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatLimitesDominio.findAllPagination. . .");  
 response = businessCatLimitesDominio.allPagination(dom);  
 LOG.info("invoke RESTCatLimitesDominio.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatLimitesDominio.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatLimitesDominio.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatLimitesDominio save(@RequestBody DominioCatLimitesDominio dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatLimitesDominio.save. . .");  
 LOG.info("dom.getLimitesDominioId(): "+dom.getLimitesDominioId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatLimitesDominio)businessCatLimitesDominio.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatLimitesDominio.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatLimitesDominio update(@RequestBody DominioCatLimitesDominio dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatLimitesDominio.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatLimitesDominio)businessCatLimitesDominio.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatLimitesDominio.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatLimitesDominio delete(@RequestBody DominioCatLimitesDominio dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatLimitesDominio.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatLimitesDominio)businessCatLimitesDominio.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatLimitesDominio.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatLimitesDominio findById(@RequestBody DominioCatLimitesDominio dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatLimitesDominio.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatLimitesDominio)businessCatLimitesDominio.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatLimitesDominio.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatLimitesDominio count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatLimitesDominio.count. . .");  
 DominioCatLimitesDominio dom = new DominioCatLimitesDominio();  
 dom.setCount(businessCatLimitesDominio.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatLimitesDominio.count: "+ex);  
 DominioCatLimitesDominio dom = new DominioCatLimitesDominio();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatLimitesDominio existsById(@RequestBody DominioCatLimitesDominio dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatLimitesDominio.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatLimitesDominio)businessCatLimitesDominio.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatLimitesDominio.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatMedidor.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatMedidor;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatMedidor;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatMedidor")  
public class RESTCatMedidor {  
 private static final Logger LOG = LogManager.getLogger(RESTCatMedidor.class);  
 @Autowired  
 private IfaceCatMedidor businessCatMedidor;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatMedidor> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatMedidor.findAll. . .");  
 List<DominioCatMedidor> lista = businessCatMedidor.findAll();  
 LOG.info("invoke RESTCatMedidor.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidor.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatMedidor dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatMedidor.findAllPagination. . .");  
 response = businessCatMedidor.allPagination(dom);  
 LOG.info("invoke RESTCatMedidor.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidor.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatMedidor.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMedidor save(@RequestBody DominioCatMedidor dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMedidor.save. . .");  
 LOG.info("dom.getMedidorId(): "+dom.getMedidorId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMedidor)businessCatMedidor.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidor.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMedidor update(@RequestBody DominioCatMedidor dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMedidor.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMedidor)businessCatMedidor.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidor.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMedidor delete(@RequestBody DominioCatMedidor dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMedidor.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMedidor)businessCatMedidor.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidor.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMedidor findById(@RequestBody DominioCatMedidor dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMedidor.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMedidor)businessCatMedidor.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidor.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMedidor count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatMedidor.count. . .");  
 DominioCatMedidor dom = new DominioCatMedidor();  
 dom.setCount(businessCatMedidor.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidor.count: "+ex);  
 DominioCatMedidor dom = new DominioCatMedidor();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMedidor existsById(@RequestBody DominioCatMedidor dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMedidor.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatMedidor)businessCatMedidor.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidor.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatMedidores.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatMedidores;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatMedidores;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatMedidores")  
public class RESTCatMedidores {  
 private static final Logger LOG = LogManager.getLogger(RESTCatMedidores.class);  
 @Autowired  
 private IfaceCatMedidores businessCatMedidores;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatMedidores> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatMedidores.findAll. . .");  
 List<DominioCatMedidores> lista = businessCatMedidores.findAll();  
 LOG.info("invoke RESTCatMedidores.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidores.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatMedidores dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatMedidores.findAllPagination. . .");  
 response = businessCatMedidores.allPagination(dom);  
 LOG.info("invoke RESTCatMedidores.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidores.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatMedidores.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMedidores save(@RequestBody DominioCatMedidores dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMedidores.save. . .");  
 LOG.info("dom.getMedidorId(): "+dom.getMedidorId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMedidores)businessCatMedidores.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidores.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMedidores update(@RequestBody DominioCatMedidores dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMedidores.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMedidores)businessCatMedidores.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidores.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMedidores delete(@RequestBody DominioCatMedidores dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMedidores.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMedidores)businessCatMedidores.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidores.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMedidores findById(@RequestBody DominioCatMedidores dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMedidores.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMedidores)businessCatMedidores.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidores.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMedidores count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatMedidores.count. . .");  
 DominioCatMedidores dom = new DominioCatMedidores();  
 dom.setCount(businessCatMedidores.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidores.count: "+ex);  
 DominioCatMedidores dom = new DominioCatMedidores();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMedidores existsById(@RequestBody DominioCatMedidores dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMedidores.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatMedidores)businessCatMedidores.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMedidores.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatMesCargo.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatMesCargo;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatMesCargo;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatMesCargo")  
public class RESTCatMesCargo {  
 private static final Logger LOG = LogManager.getLogger(RESTCatMesCargo.class);  
 @Autowired  
 private IfaceCatMesCargo businessCatMesCargo;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatMesCargo> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesCargo.findAll. . .");  
 List<DominioCatMesCargo> lista = businessCatMesCargo.findAll();  
 LOG.info("invoke RESTCatMesCargo.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesCargo.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatMesCargo dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatMesCargo.findAllPagination. . .");  
 response = businessCatMesCargo.allPagination(dom);  
 LOG.info("invoke RESTCatMesCargo.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesCargo.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatMesCargo.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesCargo save(@RequestBody DominioCatMesCargo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesCargo.save. . .");  
 LOG.info("dom.getMesId(): "+dom.getMesId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMesCargo)businessCatMesCargo.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesCargo.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesCargo update(@RequestBody DominioCatMesCargo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesCargo.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMesCargo)businessCatMesCargo.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesCargo.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesCargo delete(@RequestBody DominioCatMesCargo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesCargo.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMesCargo)businessCatMesCargo.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesCargo.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesCargo findById(@RequestBody DominioCatMesCargo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesCargo.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMesCargo)businessCatMesCargo.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesCargo.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesCargo count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesCargo.count. . .");  
 DominioCatMesCargo dom = new DominioCatMesCargo();  
 dom.setCount(businessCatMesCargo.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesCargo.count: "+ex);  
 DominioCatMesCargo dom = new DominioCatMesCargo();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesCargo existsById(@RequestBody DominioCatMesCargo dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesCargo.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatMesCargo)businessCatMesCargo.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesCargo.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatMesDia.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatMesDia;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatMesDia;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatMesDia")  
public class RESTCatMesDia {  
 private static final Logger LOG = LogManager.getLogger(RESTCatMesDia.class);  
 @Autowired  
 private IfaceCatMesDia businessCatMesDia;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatMesDia> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesDia.findAll. . .");  
 List<DominioCatMesDia> lista = businessCatMesDia.findAll();  
 LOG.info("invoke RESTCatMesDia.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDia.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatMesDia dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatMesDia.findAllPagination. . .");  
 response = businessCatMesDia.allPagination(dom);  
 LOG.info("invoke RESTCatMesDia.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDia.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatMesDia.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesDia save(@RequestBody DominioCatMesDia dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesDia.save. . .");  
 LOG.info("dom.getMesDiaId(): "+dom.getMesDiaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMesDia)businessCatMesDia.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDia.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesDia update(@RequestBody DominioCatMesDia dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesDia.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMesDia)businessCatMesDia.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDia.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesDia delete(@RequestBody DominioCatMesDia dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesDia.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMesDia)businessCatMesDia.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDia.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesDia findById(@RequestBody DominioCatMesDia dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesDia.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMesDia)businessCatMesDia.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDia.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesDia count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesDia.count. . .");  
 DominioCatMesDia dom = new DominioCatMesDia();  
 dom.setCount(businessCatMesDia.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDia.count: "+ex);  
 DominioCatMesDia dom = new DominioCatMesDia();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesDia existsById(@RequestBody DominioCatMesDia dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesDia.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatMesDia)businessCatMesDia.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDia.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatMesDiaFirma.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatMesDiaFirma;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatMesDiaFirma;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatMesDiaFirma")  
public class RESTCatMesDiaFirma {  
 private static final Logger LOG = LogManager.getLogger(RESTCatMesDiaFirma.class);  
 @Autowired  
 private IfaceCatMesDiaFirma businessCatMesDiaFirma;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatMesDiaFirma> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesDiaFirma.findAll. . .");  
 List<DominioCatMesDiaFirma> lista = businessCatMesDiaFirma.findAll();  
 LOG.info("invoke RESTCatMesDiaFirma.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDiaFirma.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatMesDiaFirma dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatMesDiaFirma.findAllPagination. . .");  
 response = businessCatMesDiaFirma.allPagination(dom);  
 LOG.info("invoke RESTCatMesDiaFirma.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDiaFirma.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatMesDiaFirma.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesDiaFirma save(@RequestBody DominioCatMesDiaFirma dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesDiaFirma.save. . .");  
 LOG.info("dom.getMesDiaFirmaId(): "+dom.getMesDiaFirmaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMesDiaFirma)businessCatMesDiaFirma.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDiaFirma.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesDiaFirma update(@RequestBody DominioCatMesDiaFirma dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesDiaFirma.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMesDiaFirma)businessCatMesDiaFirma.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDiaFirma.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesDiaFirma delete(@RequestBody DominioCatMesDiaFirma dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesDiaFirma.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMesDiaFirma)businessCatMesDiaFirma.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDiaFirma.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesDiaFirma findById(@RequestBody DominioCatMesDiaFirma dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesDiaFirma.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMesDiaFirma)businessCatMesDiaFirma.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDiaFirma.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesDiaFirma count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesDiaFirma.count. . .");  
 DominioCatMesDiaFirma dom = new DominioCatMesDiaFirma();  
 dom.setCount(businessCatMesDiaFirma.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDiaFirma.count: "+ex);  
 DominioCatMesDiaFirma dom = new DominioCatMesDiaFirma();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMesDiaFirma existsById(@RequestBody DominioCatMesDiaFirma dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMesDiaFirma.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatMesDiaFirma)businessCatMesDiaFirma.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMesDiaFirma.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatMetodoPago.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatMetodoPago;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatMetodoPago;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatMetodoPago")  
public class RESTCatMetodoPago {  
 private static final Logger LOG = LogManager.getLogger(RESTCatMetodoPago.class);  
 @Autowired  
 private IfaceCatMetodoPago businessCatMetodoPago;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatMetodoPago> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatMetodoPago.findAll. . .");  
 List<DominioCatMetodoPago> lista = businessCatMetodoPago.findAll();  
 LOG.info("invoke RESTCatMetodoPago.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMetodoPago.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatMetodoPago dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatMetodoPago.findAllPagination. . .");  
 response = businessCatMetodoPago.allPagination(dom);  
 LOG.info("invoke RESTCatMetodoPago.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMetodoPago.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatMetodoPago.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMetodoPago save(@RequestBody DominioCatMetodoPago dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMetodoPago.save. . .");  
 LOG.info("dom.getMetodoPagoId(): "+dom.getMetodoPagoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMetodoPago)businessCatMetodoPago.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMetodoPago.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMetodoPago update(@RequestBody DominioCatMetodoPago dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMetodoPago.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMetodoPago)businessCatMetodoPago.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMetodoPago.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMetodoPago delete(@RequestBody DominioCatMetodoPago dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMetodoPago.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMetodoPago)businessCatMetodoPago.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMetodoPago.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMetodoPago findById(@RequestBody DominioCatMetodoPago dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMetodoPago.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMetodoPago)businessCatMetodoPago.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMetodoPago.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMetodoPago count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatMetodoPago.count. . .");  
 DominioCatMetodoPago dom = new DominioCatMetodoPago();  
 dom.setCount(businessCatMetodoPago.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMetodoPago.count: "+ex);  
 DominioCatMetodoPago dom = new DominioCatMetodoPago();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMetodoPago existsById(@RequestBody DominioCatMetodoPago dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMetodoPago.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatMetodoPago)businessCatMetodoPago.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMetodoPago.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatMonedas.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatMonedas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatMonedas;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatMonedas")  
public class RESTCatMonedas {  
 private static final Logger LOG = LogManager.getLogger(RESTCatMonedas.class);  
 @Autowired  
 private IfaceCatMonedas businessCatMonedas;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatMonedas> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatMonedas.findAll. . .");  
 List<DominioCatMonedas> lista = businessCatMonedas.findAll();  
 LOG.info("invoke RESTCatMonedas.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMonedas.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatMonedas dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatMonedas.findAllPagination. . .");  
 response = businessCatMonedas.allPagination(dom);  
 LOG.info("invoke RESTCatMonedas.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMonedas.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatMonedas.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMonedas save(@RequestBody DominioCatMonedas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMonedas.save. . .");  
 LOG.info("dom.getMonedaId(): "+dom.getMonedaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMonedas)businessCatMonedas.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMonedas.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMonedas update(@RequestBody DominioCatMonedas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMonedas.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMonedas)businessCatMonedas.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMonedas.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMonedas delete(@RequestBody DominioCatMonedas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMonedas.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMonedas)businessCatMonedas.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMonedas.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMonedas findById(@RequestBody DominioCatMonedas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMonedas.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatMonedas)businessCatMonedas.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMonedas.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMonedas count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatMonedas.count. . .");  
 DominioCatMonedas dom = new DominioCatMonedas();  
 dom.setCount(businessCatMonedas.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMonedas.count: "+ex);  
 DominioCatMonedas dom = new DominioCatMonedas();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatMonedas existsById(@RequestBody DominioCatMonedas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatMonedas.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatMonedas)businessCatMonedas.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatMonedas.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatOpcionPlanta.java

package mx.mbi.dgc.microservices.controller;  
  
import java.math.BigDecimal;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatOpcionPlanta;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatOpcionPlanta;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceCatDetalleOpciones;  
import mx.mbi.dgc.dominio.DominioCatDetalleOpciones;  
  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatOpcionPlanta")  
public class RESTCatOpcionPlanta {  
 private static final Logger LOG = LogManager.getLogger(RESTCatOpcionPlanta.class);  
 @Autowired  
 private IfaceCatOpcionPlanta businessCatOpcionPlanta;  
 @Autowired  
 private IfaceCatDetalleOpciones businessCatDetalleOpciones;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatOpcionPlanta> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatOpcionPlanta.findAll. . .");  
 List<DominioCatOpcionPlanta> lista = businessCatOpcionPlanta.findAll();  
 LOG.info("invoke RESTCatOpcionPlanta.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatOpcionPlanta.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatOpcionPlanta dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatOpcionPlanta.findAllPagination. . .");  
 response = businessCatOpcionPlanta.allPagination(dom);  
 LOG.info("invoke RESTCatOpcionPlanta.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatOpcionPlanta.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatOpcionPlanta.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatOpcionPlanta save(@RequestBody DominioCatOpcionPlanta dom) throws Exception {  
 try {  
 DominioCatDetalleOpciones dom2 = new DominioCatDetalleOpciones();  
 JRUtil.copyProperties(dom2, dom.getDetalleOpcionesId());  
 long id = businessCatDetalleOpciones.count();  
 id = id+5;  
 dom2.setDetalleOpcionesId(BigDecimal.valueOf(id));  
 long id2 = businessCatOpcionPlanta.count();  
 id2 = id2+6;  
 dom.setOpcionPlantaId(BigDecimal.valueOf(id2));  
 dom.getDetalleOpcionesId().setDetalleOpcionesId(BigDecimal.valueOf(id));  
 LOG.info("invoke RESTCatOpcionPlanta.save. . .");  
 LOG.info("dom.getOpcionPlantaId(): "+dom.getOpcionPlantaId());  
 System.out.println("OpcionPlantaId" + dom.getOpcionPlantaId() +dom.getDetalleOpcionesId().getDetalleOpcionesId() + dom.getPlantaId().getPlantaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom2 = (DominioCatDetalleOpciones) businessCatDetalleOpciones.save(dom2);  
 dom = (DominioCatOpcionPlanta)businessCatOpcionPlanta.save(dom);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatOpcionPlanta.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatOpcionPlanta update(@RequestBody DominioCatOpcionPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatOpcionPlanta.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatOpcionPlanta)businessCatOpcionPlanta.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatOpcionPlanta.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatOpcionPlanta delete(@RequestBody DominioCatOpcionPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatOpcionPlanta.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = businessCatOpcionPlanta.findByOpcionPlanta(dom);  
 return (DominioCatOpcionPlanta)businessCatOpcionPlanta.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatOpcionPlanta.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatOpcionPlanta findById(@RequestBody DominioCatOpcionPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatOpcionPlanta.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatOpcionPlanta)businessCatOpcionPlanta.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatOpcionPlanta.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatOpcionPlanta count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatOpcionPlanta.count. . .");  
 DominioCatOpcionPlanta dom = new DominioCatOpcionPlanta();  
 dom.setCount(businessCatOpcionPlanta.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatOpcionPlanta.count: "+ex);  
 DominioCatOpcionPlanta dom = new DominioCatOpcionPlanta();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatOpcionPlanta existsById(@RequestBody DominioCatOpcionPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatOpcionPlanta.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatOpcionPlanta)businessCatOpcionPlanta.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatOpcionPlanta.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatPaises.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatPaises;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatPaises;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatPaises")  
public class RESTCatPaises {  
 private static final Logger LOG = LogManager.getLogger(RESTCatPaises.class);  
 @Autowired  
 private IfaceCatPaises businessCatPaises;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatPaises> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatPaises.findAll. . .");  
 List<DominioCatPaises> lista = businessCatPaises.findAll();  
 LOG.info("invoke RESTCatPaises.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPaises.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatPaises dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatPaises.findAllPagination. . .");  
 response = businessCatPaises.allPagination(dom);  
 LOG.info("invoke RESTCatPaises.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPaises.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatPaises.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPaises save(@RequestBody DominioCatPaises dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPaises.save. . .");  
 LOG.info("dom.getPaisId(): "+dom.getPaisId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPaises)businessCatPaises.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPaises.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPaises update(@RequestBody DominioCatPaises dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPaises.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPaises)businessCatPaises.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPaises.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPaises delete(@RequestBody DominioCatPaises dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPaises.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPaises)businessCatPaises.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPaises.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPaises findById(@RequestBody DominioCatPaises dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPaises.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPaises)businessCatPaises.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPaises.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPaises count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatPaises.count. . .");  
 DominioCatPaises dom = new DominioCatPaises();  
 dom.setCount(businessCatPaises.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPaises.count: "+ex);  
 DominioCatPaises dom = new DominioCatPaises();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPaises existsById(@RequestBody DominioCatPaises dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPaises.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatPaises)businessCatPaises.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPaises.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatPermiso.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatPermiso;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatPermiso;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatPermiso")  
public class RESTCatPermiso {  
 private static final Logger LOG = LogManager.getLogger(RESTCatPermiso.class);  
 @Autowired  
 private IfaceCatPermiso businessCatPermiso;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatPermiso> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatPermiso.findAll. . .");  
 List<DominioCatPermiso> lista = businessCatPermiso.findAll();  
 LOG.info("invoke RESTCatPermiso.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPermiso.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatPermiso dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatPermiso.findAllPagination. . .");  
 response = businessCatPermiso.allPagination(dom);  
 LOG.info("invoke RESTCatPermiso.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPermiso.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatPermiso.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPermiso save(@RequestBody DominioCatPermiso dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPermiso.save. . .");  
 LOG.info("dom.getPermisoId(): "+dom.getPermisoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPermiso)businessCatPermiso.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPermiso.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPermiso update(@RequestBody DominioCatPermiso dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPermiso.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPermiso)businessCatPermiso.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPermiso.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPermiso delete(@RequestBody DominioCatPermiso dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPermiso.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPermiso)businessCatPermiso.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPermiso.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPermiso findById(@RequestBody DominioCatPermiso dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPermiso.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPermiso)businessCatPermiso.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPermiso.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPermiso count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatPermiso.count. . .");  
 DominioCatPermiso dom = new DominioCatPermiso();  
 dom.setCount(businessCatPermiso.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPermiso.count: "+ex);  
 DominioCatPermiso dom = new DominioCatPermiso();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPermiso existsById(@RequestBody DominioCatPermiso dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPermiso.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatPermiso)businessCatPermiso.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPermiso.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatPlanta.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatPlanta;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatPlanta;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatPlanta")  
public class RESTCatPlanta {  
 private static final Logger LOG = LogManager.getLogger(RESTCatPlanta.class);  
 @Autowired  
 private IfaceCatPlanta businessCatPlanta;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatPlanta> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlanta.findAll. . .");  
 List<DominioCatPlanta> lista = businessCatPlanta.findAll();  
 LOG.info("invoke RESTCatPlanta.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlanta.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatPlanta dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatPlanta.findAllPagination. . .");  
 response = businessCatPlanta.allPagination(dom);  
 LOG.info("invoke RESTCatPlanta.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlanta.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatPlanta.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPlanta save(@RequestBody DominioCatPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlanta.save. . .");  
 LOG.info("dom.getPlantaId(): "+dom.getPlantaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPlanta)businessCatPlanta.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlanta.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPlanta update(@RequestBody DominioCatPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlanta.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPlanta)businessCatPlanta.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlanta.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPlanta delete(@RequestBody DominioCatPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlanta.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPlanta)businessCatPlanta.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlanta.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPlanta findById(@RequestBody DominioCatPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlanta.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPlanta)businessCatPlanta.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlanta.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPlanta count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlanta.count. . .");  
 DominioCatPlanta dom = new DominioCatPlanta();  
 dom.setCount(businessCatPlanta.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlanta.count: "+ex);  
 DominioCatPlanta dom = new DominioCatPlanta();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPlanta existsById(@RequestBody DominioCatPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlanta.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatPlanta)businessCatPlanta.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlanta.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatPlantaTipoFacturas.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatPlantaTipoFacturas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatPlantaTipoFacturas;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatPlantaTipoFacturas")  
public class RESTCatPlantaTipoFacturas {  
 private static final Logger LOG = LogManager.getLogger(RESTCatPlantaTipoFacturas.class);  
 @Autowired  
 private IfaceCatPlantaTipoFacturas businessCatPlantaTipoFacturas;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatPlantaTipoFacturas> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlantaTipoFacturas.findAll. . .");  
 List<DominioCatPlantaTipoFacturas> lista = businessCatPlantaTipoFacturas.findAll();  
 LOG.info("invoke RESTCatPlantaTipoFacturas.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlantaTipoFacturas.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatPlantaTipoFacturas dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatPlantaTipoFacturas.findAllPagination. . .");  
 response = businessCatPlantaTipoFacturas.allPagination(dom);  
 LOG.info("invoke RESTCatPlantaTipoFacturas.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlantaTipoFacturas.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatPlantaTipoFacturas.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPlantaTipoFacturas save(@RequestBody DominioCatPlantaTipoFacturas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlantaTipoFacturas.save. . .");  
 LOG.info("dom.getPlantaTipoFacturaId(): "+dom.getPlantaTipoFacturaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPlantaTipoFacturas)businessCatPlantaTipoFacturas.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlantaTipoFacturas.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPlantaTipoFacturas update(@RequestBody DominioCatPlantaTipoFacturas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlantaTipoFacturas.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPlantaTipoFacturas)businessCatPlantaTipoFacturas.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlantaTipoFacturas.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPlantaTipoFacturas delete(@RequestBody DominioCatPlantaTipoFacturas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlantaTipoFacturas.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPlantaTipoFacturas)businessCatPlantaTipoFacturas.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlantaTipoFacturas.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPlantaTipoFacturas findById(@RequestBody DominioCatPlantaTipoFacturas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlantaTipoFacturas.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPlantaTipoFacturas)businessCatPlantaTipoFacturas.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlantaTipoFacturas.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPlantaTipoFacturas count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlantaTipoFacturas.count. . .");  
 DominioCatPlantaTipoFacturas dom = new DominioCatPlantaTipoFacturas();  
 dom.setCount(businessCatPlantaTipoFacturas.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlantaTipoFacturas.count: "+ex);  
 DominioCatPlantaTipoFacturas dom = new DominioCatPlantaTipoFacturas();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPlantaTipoFacturas existsById(@RequestBody DominioCatPlantaTipoFacturas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPlantaTipoFacturas.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatPlantaTipoFacturas)businessCatPlantaTipoFacturas.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPlantaTipoFacturas.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatPropCincominutal.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatPropCincominutal;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatPropCincominutal;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatPropCincominutal")  
public class RESTCatPropCincominutal {  
 private static final Logger LOG = LogManager.getLogger(RESTCatPropCincominutal.class);  
 @Autowired  
 private IfaceCatPropCincominutal businessCatPropCincominutal;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatPropCincominutal> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatPropCincominutal.findAll. . .");  
 List<DominioCatPropCincominutal> lista = businessCatPropCincominutal.findAll();  
 LOG.info("invoke RESTCatPropCincominutal.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPropCincominutal.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatPropCincominutal dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatPropCincominutal.findAllPagination. . .");  
 response = businessCatPropCincominutal.allPagination(dom);  
 LOG.info("invoke RESTCatPropCincominutal.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPropCincominutal.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatPropCincominutal.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPropCincominutal save(@RequestBody DominioCatPropCincominutal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPropCincominutal.save. . .");  
 LOG.info("dom.getPropiedadId(): "+dom.getPropiedadId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPropCincominutal)businessCatPropCincominutal.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPropCincominutal.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPropCincominutal update(@RequestBody DominioCatPropCincominutal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPropCincominutal.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPropCincominutal)businessCatPropCincominutal.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPropCincominutal.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPropCincominutal delete(@RequestBody DominioCatPropCincominutal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPropCincominutal.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPropCincominutal)businessCatPropCincominutal.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPropCincominutal.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPropCincominutal findById(@RequestBody DominioCatPropCincominutal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPropCincominutal.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatPropCincominutal)businessCatPropCincominutal.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPropCincominutal.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPropCincominutal count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatPropCincominutal.count. . .");  
 DominioCatPropCincominutal dom = new DominioCatPropCincominutal();  
 dom.setCount(businessCatPropCincominutal.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPropCincominutal.count: "+ex);  
 DominioCatPropCincominutal dom = new DominioCatPropCincominutal();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatPropCincominutal existsById(@RequestBody DominioCatPropCincominutal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatPropCincominutal.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatPropCincominutal)businessCatPropCincominutal.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatPropCincominutal.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatQuery.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatQuery;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatQuery;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatQuery")  
public class RESTCatQuery {  
 private static final Logger LOG = LogManager.getLogger(RESTCatQuery.class);  
 @Autowired  
 private IfaceCatQuery businessCatQuery;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatQuery> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatQuery.findAll. . .");  
 List<DominioCatQuery> lista = businessCatQuery.findAll();  
 LOG.info("invoke RESTCatQuery.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatQuery.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatQuery dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatQuery.findAllPagination. . .");  
 response = businessCatQuery.allPagination(dom);  
 LOG.info("invoke RESTCatQuery.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatQuery.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatQuery.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatQuery save(@RequestBody DominioCatQuery dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatQuery.save. . .");  
 LOG.info("dom.getQueryId(): "+dom.getQueryId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatQuery)businessCatQuery.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatQuery.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatQuery update(@RequestBody DominioCatQuery dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatQuery.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatQuery)businessCatQuery.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatQuery.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatQuery delete(@RequestBody DominioCatQuery dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatQuery.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatQuery)businessCatQuery.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatQuery.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatQuery findById(@RequestBody DominioCatQuery dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatQuery.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatQuery)businessCatQuery.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatQuery.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatQuery count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatQuery.count. . .");  
 DominioCatQuery dom = new DominioCatQuery();  
 dom.setCount(businessCatQuery.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatQuery.count: "+ex);  
 DominioCatQuery dom = new DominioCatQuery();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatQuery existsById(@RequestBody DominioCatQuery dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatQuery.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatQuery)businessCatQuery.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatQuery.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatRangoVariable.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatRangoVariable;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatRangoVariable;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatRangoVariable")  
public class RESTCatRangoVariable {  
 private static final Logger LOG = LogManager.getLogger(RESTCatRangoVariable.class);  
 @Autowired  
 private IfaceCatRangoVariable businessCatRangoVariable;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatRangoVariable> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatRangoVariable.findAll. . .");  
 List<DominioCatRangoVariable> lista = businessCatRangoVariable.findAll();  
 LOG.info("invoke RESTCatRangoVariable.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRangoVariable.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatRangoVariable dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatRangoVariable.findAllPagination. . .");  
 response = businessCatRangoVariable.allPagination(dom);  
 LOG.info("invoke RESTCatRangoVariable.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRangoVariable.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatRangoVariable.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRangoVariable save(@RequestBody DominioCatRangoVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatRangoVariable.save. . .");  
 LOG.info("dom.getRangoId(): "+dom.getRangoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatRangoVariable)businessCatRangoVariable.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRangoVariable.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRangoVariable update(@RequestBody DominioCatRangoVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatRangoVariable.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatRangoVariable)businessCatRangoVariable.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRangoVariable.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRangoVariable delete(@RequestBody DominioCatRangoVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatRangoVariable.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatRangoVariable)businessCatRangoVariable.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRangoVariable.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRangoVariable findById(@RequestBody DominioCatRangoVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatRangoVariable.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatRangoVariable)businessCatRangoVariable.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRangoVariable.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRangoVariable count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatRangoVariable.count. . .");  
 DominioCatRangoVariable dom = new DominioCatRangoVariable();  
 dom.setCount(businessCatRangoVariable.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRangoVariable.count: "+ex);  
 DominioCatRangoVariable dom = new DominioCatRangoVariable();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRangoVariable existsById(@RequestBody DominioCatRangoVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatRangoVariable.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatRangoVariable)businessCatRangoVariable.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRangoVariable.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatRegimenFiscal.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatRegimenFiscal;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatRegimenFiscal;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatRegimenFiscal")  
public class RESTCatRegimenFiscal {  
 private static final Logger LOG = LogManager.getLogger(RESTCatRegimenFiscal.class);  
 @Autowired  
 private IfaceCatRegimenFiscal businessCatRegimenFiscal;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatRegimenFiscal> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatRegimenFiscal.findAll. . .");  
 List<DominioCatRegimenFiscal> lista = businessCatRegimenFiscal.findAll();  
 LOG.info("invoke RESTCatRegimenFiscal.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRegimenFiscal.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatRegimenFiscal dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatRegimenFiscal.findAllPagination. . .");  
 response = businessCatRegimenFiscal.allPagination(dom);  
 LOG.info("invoke RESTCatRegimenFiscal.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRegimenFiscal.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatRegimenFiscal.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRegimenFiscal save(@RequestBody DominioCatRegimenFiscal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatRegimenFiscal.save. . .");  
 LOG.info("dom.getRegimenFiscalId(): "+dom.getRegimenFiscalId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatRegimenFiscal)businessCatRegimenFiscal.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRegimenFiscal.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRegimenFiscal update(@RequestBody DominioCatRegimenFiscal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatRegimenFiscal.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatRegimenFiscal)businessCatRegimenFiscal.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRegimenFiscal.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRegimenFiscal delete(@RequestBody DominioCatRegimenFiscal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatRegimenFiscal.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatRegimenFiscal)businessCatRegimenFiscal.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRegimenFiscal.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRegimenFiscal findById(@RequestBody DominioCatRegimenFiscal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatRegimenFiscal.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatRegimenFiscal)businessCatRegimenFiscal.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRegimenFiscal.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRegimenFiscal count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatRegimenFiscal.count. . .");  
 DominioCatRegimenFiscal dom = new DominioCatRegimenFiscal();  
 dom.setCount(businessCatRegimenFiscal.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRegimenFiscal.count: "+ex);  
 DominioCatRegimenFiscal dom = new DominioCatRegimenFiscal();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRegimenFiscal existsById(@RequestBody DominioCatRegimenFiscal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatRegimenFiscal.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatRegimenFiscal)businessCatRegimenFiscal.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRegimenFiscal.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatReglaNegocio.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatReglaNegocio;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatReglaNegocio;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatReglaNegocio")  
public class RESTCatReglaNegocio {  
 private static final Logger LOG = LogManager.getLogger(RESTCatReglaNegocio.class);  
 @Autowired  
 private IfaceCatReglaNegocio businessCatReglaNegocio;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatReglaNegocio> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatReglaNegocio.findAll. . .");  
 List<DominioCatReglaNegocio> lista = businessCatReglaNegocio.findAll();  
 LOG.info("invoke RESTCatReglaNegocio.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatReglaNegocio.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatReglaNegocio dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatReglaNegocio.findAllPagination. . .");  
 response = businessCatReglaNegocio.allPagination(dom);  
 LOG.info("invoke RESTCatReglaNegocio.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatReglaNegocio.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatReglaNegocio.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatReglaNegocio save(@RequestBody DominioCatReglaNegocio dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatReglaNegocio.save. . .");  
 LOG.info("dom.getReglaNegocioId(): "+dom.getReglaNegocioId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatReglaNegocio)businessCatReglaNegocio.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatReglaNegocio.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatReglaNegocio update(@RequestBody DominioCatReglaNegocio dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatReglaNegocio.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatReglaNegocio)businessCatReglaNegocio.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatReglaNegocio.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatReglaNegocio delete(@RequestBody DominioCatReglaNegocio dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatReglaNegocio.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatReglaNegocio)businessCatReglaNegocio.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatReglaNegocio.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatReglaNegocio findById(@RequestBody DominioCatReglaNegocio dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatReglaNegocio.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatReglaNegocio)businessCatReglaNegocio.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatReglaNegocio.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatReglaNegocio count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatReglaNegocio.count. . .");  
 DominioCatReglaNegocio dom = new DominioCatReglaNegocio();  
 dom.setCount(businessCatReglaNegocio.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatReglaNegocio.count: "+ex);  
 DominioCatReglaNegocio dom = new DominioCatReglaNegocio();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatReglaNegocio existsById(@RequestBody DominioCatReglaNegocio dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatReglaNegocio.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatReglaNegocio)businessCatReglaNegocio.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatReglaNegocio.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatRol.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatRol;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatRol;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatRol")  
public class RESTCatRol {  
 private static final Logger LOG = LogManager.getLogger(RESTCatRol.class);  
 @Autowired  
 private IfaceCatRol businessCatRol;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatRol> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatRol.findAll. . .");  
 List<DominioCatRol> lista = businessCatRol.findAll();  
 LOG.info("invoke RESTCatRol.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRol.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatRol dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatRol.findAllPagination. . .");  
 response = businessCatRol.allPagination(dom);  
 LOG.info("invoke RESTCatRol.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRol.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatRol.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRol save(@RequestBody DominioCatRol dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatRol.save. . .");  
 LOG.info("dom.getRolId(): "+dom.getRolId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatRol)businessCatRol.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRol.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRol update(@RequestBody DominioCatRol dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatRol.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatRol)businessCatRol.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRol.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRol delete(@RequestBody DominioCatRol dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatRol.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatRol)businessCatRol.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRol.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRol findById(@RequestBody DominioCatRol dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatRol.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatRol)businessCatRol.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRol.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRol count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatRol.count. . .");  
 DominioCatRol dom = new DominioCatRol();  
 dom.setCount(businessCatRol.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRol.count: "+ex);  
 DominioCatRol dom = new DominioCatRol();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatRol existsById(@RequestBody DominioCatRol dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatRol.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatRol)businessCatRol.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatRol.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatServicioImpuestos.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatServicioImpuestos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatServicioImpuestos;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatServicioImpuestos")  
public class RESTCatServicioImpuestos {  
 private static final Logger LOG = LogManager.getLogger(RESTCatServicioImpuestos.class);  
 @Autowired  
 private IfaceCatServicioImpuestos businessCatServicioImpuestos;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatServicioImpuestos> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatServicioImpuestos.findAll. . .");  
 List<DominioCatServicioImpuestos> lista = businessCatServicioImpuestos.findAll();  
 LOG.info("invoke RESTCatServicioImpuestos.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicioImpuestos.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatServicioImpuestos dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatServicioImpuestos.findAllPagination. . .");  
 response = businessCatServicioImpuestos.allPagination(dom);  
 LOG.info("invoke RESTCatServicioImpuestos.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicioImpuestos.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatServicioImpuestos.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatServicioImpuestos save(@RequestBody DominioCatServicioImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatServicioImpuestos.save. . .");  
 LOG.info("dom.getServicioImpuestoId(): "+dom.getServicioImpuestoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatServicioImpuestos)businessCatServicioImpuestos.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicioImpuestos.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatServicioImpuestos update(@RequestBody DominioCatServicioImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatServicioImpuestos.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatServicioImpuestos)businessCatServicioImpuestos.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicioImpuestos.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatServicioImpuestos delete(@RequestBody DominioCatServicioImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatServicioImpuestos.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatServicioImpuestos)businessCatServicioImpuestos.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicioImpuestos.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatServicioImpuestos findById(@RequestBody DominioCatServicioImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatServicioImpuestos.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatServicioImpuestos)businessCatServicioImpuestos.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicioImpuestos.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatServicioImpuestos count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatServicioImpuestos.count. . .");  
 DominioCatServicioImpuestos dom = new DominioCatServicioImpuestos();  
 dom.setCount(businessCatServicioImpuestos.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicioImpuestos.count: "+ex);  
 DominioCatServicioImpuestos dom = new DominioCatServicioImpuestos();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatServicioImpuestos existsById(@RequestBody DominioCatServicioImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatServicioImpuestos.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatServicioImpuestos)businessCatServicioImpuestos.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicioImpuestos.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatServicios.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatServicios;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatServicios;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatServicios")  
public class RESTCatServicios {  
 private static final Logger LOG = LogManager.getLogger(RESTCatServicios.class);  
 @Autowired  
 private IfaceCatServicios businessCatServicios;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatServicios> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatServicios.findAll. . .");  
 List<DominioCatServicios> lista = businessCatServicios.findAll();  
 LOG.info("invoke RESTCatServicios.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicios.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatServicios dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatServicios.findAllPagination. . .");  
 response = businessCatServicios.allPagination(dom);  
 LOG.info("invoke RESTCatServicios.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicios.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatServicios.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatServicios save(@RequestBody DominioCatServicios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatServicios.save. . .");  
 LOG.info("dom.getServicioId(): "+dom.getServicioId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatServicios)businessCatServicios.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicios.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatServicios update(@RequestBody DominioCatServicios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatServicios.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatServicios)businessCatServicios.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicios.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatServicios delete(@RequestBody DominioCatServicios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatServicios.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatServicios)businessCatServicios.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicios.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatServicios findById(@RequestBody DominioCatServicios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatServicios.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatServicios)businessCatServicios.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicios.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatServicios count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatServicios.count. . .");  
 DominioCatServicios dom = new DominioCatServicios();  
 dom.setCount(businessCatServicios.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicios.count: "+ex);  
 DominioCatServicios dom = new DominioCatServicios();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatServicios existsById(@RequestBody DominioCatServicios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatServicios.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatServicios)businessCatServicios.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatServicios.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatSistemas.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatSistemas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatSistemas;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatSistemas")  
public class RESTCatSistemas {  
 private static final Logger LOG = LogManager.getLogger(RESTCatSistemas.class);  
 @Autowired  
 private IfaceCatSistemas businessCatSistemas;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatSistemas> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatSistemas.findAll. . .");  
 List<DominioCatSistemas> lista = businessCatSistemas.findAll();  
 LOG.info("invoke RESTCatSistemas.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSistemas.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatSistemas dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatSistemas.findAllPagination. . .");  
 response = businessCatSistemas.allPagination(dom);  
 LOG.info("invoke RESTCatSistemas.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSistemas.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatSistemas.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatSistemas save(@RequestBody DominioCatSistemas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatSistemas.save. . .");  
 LOG.info("dom.getSistemaId(): "+dom.getSistemaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatSistemas)businessCatSistemas.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSistemas.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatSistemas update(@RequestBody DominioCatSistemas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatSistemas.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatSistemas)businessCatSistemas.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSistemas.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatSistemas delete(@RequestBody DominioCatSistemas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatSistemas.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatSistemas)businessCatSistemas.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSistemas.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatSistemas findById(@RequestBody DominioCatSistemas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatSistemas.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatSistemas)businessCatSistemas.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSistemas.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatSistemas count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatSistemas.count. . .");  
 DominioCatSistemas dom = new DominioCatSistemas();  
 dom.setCount(businessCatSistemas.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSistemas.count: "+ex);  
 DominioCatSistemas dom = new DominioCatSistemas();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatSistemas existsById(@RequestBody DominioCatSistemas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatSistemas.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatSistemas)businessCatSistemas.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSistemas.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatSubCargos.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatSubCargos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatSubCargos;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatSubCargos")  
public class RESTCatSubCargos {  
 private static final Logger LOG = LogManager.getLogger(RESTCatSubCargos.class);  
 @Autowired  
 private IfaceCatSubCargos businessCatSubCargos;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatSubCargos> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatSubCargos.findAll. . .");  
 List<DominioCatSubCargos> lista = businessCatSubCargos.findAll();  
 LOG.info("invoke RESTCatSubCargos.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSubCargos.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatSubCargos dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatSubCargos.findAllPagination. . .");  
 response = businessCatSubCargos.allPagination(dom);  
 LOG.info("invoke RESTCatSubCargos.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSubCargos.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatSubCargos.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatSubCargos save(@RequestBody DominioCatSubCargos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatSubCargos.save. . .");  
 LOG.info("dom.getSubCargoId(): "+dom.getSubCargoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatSubCargos)businessCatSubCargos.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSubCargos.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatSubCargos update(@RequestBody DominioCatSubCargos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatSubCargos.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatSubCargos)businessCatSubCargos.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSubCargos.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatSubCargos delete(@RequestBody DominioCatSubCargos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatSubCargos.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatSubCargos)businessCatSubCargos.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSubCargos.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatSubCargos findById(@RequestBody DominioCatSubCargos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatSubCargos.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatSubCargos)businessCatSubCargos.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSubCargos.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatSubCargos count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatSubCargos.count. . .");  
 DominioCatSubCargos dom = new DominioCatSubCargos();  
 dom.setCount(businessCatSubCargos.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSubCargos.count: "+ex);  
 DominioCatSubCargos dom = new DominioCatSubCargos();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatSubCargos existsById(@RequestBody DominioCatSubCargos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatSubCargos.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatSubCargos)businessCatSubCargos.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatSubCargos.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatTags.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceCatTags;  
import mx.mbi.dgc.dominio.DominioCatPlanta;  
import mx.mbi.dgc.dominio.DominioCatTags;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatTags")  
public class RESTCatTags {  
 private static final Logger LOG = LogManager.getLogger(RESTCatTags.class);  
 @Autowired  
 private IfaceCatTags businessCatTags;  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAll", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioCatTags> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTags.findAll. . .");  
 List<DominioCatTags> lista = businessCatTags.findAll();  
 LOG.info("invoke RESTCatTags.findAll size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatTags.findAll: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findByPlantaId", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioCatTags> findByPlantaId(@RequestBody DominioCatPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTags.findByPlantaId. . .");  
 System.out.println("planta:"+dom.getPlantaId());  
 List<DominioCatTags> lista = businessCatTags.findByPlantaId(dom);  
 LOG.info("invoke RESTCatTags.findByPlantaId size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatTags.findByPlantaId: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAllPagination", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioRespuesta findAllPagination(@RequestBody DominioCatTags dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatTags.findAllPagination. . .");  
 response = businessCatTags.allPagination(dom);  
 LOG.info("invoke RESTCatTags.findAllPagination size lista return: " + response.getData().size());  
 return response;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatTags.findAllPagination: " + ex);  
 response.setMensaje("Error in invoke RESTCatTags.findAllPagination: " + ex);  
 return response;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/save", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatTags save(@RequestBody DominioCatTags dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTags.save. . .");  
 LOG.info("dom.getTagId(): " + dom.getTagId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTags) businessCatTags.save(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatTags.save: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/update", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatTags update(@RequestBody DominioCatTags dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTags.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTags) businessCatTags.update(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatTags.update: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/delete", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatTags delete(@RequestBody DominioCatTags dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTags.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTags) businessCatTags.delete(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatTags.delete: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatTags findById(@RequestBody DominioCatTags dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTags.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTags) businessCatTags.findById(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatTags.findById: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/count", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatTags count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTags.count. . .");  
 DominioCatTags dom = new DominioCatTags();  
 dom.setCount(businessCatTags.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatTags.count: " + ex);  
 DominioCatTags dom = new DominioCatTags();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/existsById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioCatTags existsById(@RequestBody DominioCatTags dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTags.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatTags) businessCatTags.existsById(dom);  
 if (dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTCatTags.existsById: " + ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatTipoDato.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceCatTipoDato;  
import mx.mbi.dgc.dominio.DominioCatTipoDato;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatTipoDato")  
public class RESTCatTipoDato {  
 private static final Logger LOG = LogManager.getLogger(RESTCatTipoDato.class);  
 @Autowired  
 private IfaceCatTipoDato businessCatTipoDato;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatTipoDato> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoDato.findAll. . .");  
 List<DominioCatTipoDato> lista = businessCatTipoDato.findAll();  
 LOG.info("invoke RESTCatTipoDato.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDato.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatTipoDato dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatTipoDato.findAllPagination. . .");  
 response = businessCatTipoDato.allPagination(dom);  
 LOG.info("invoke RESTCatTipoDato.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDato.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatTipoDato.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoDato save(@RequestBody DominioCatTipoDato dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoDato.save. . .");  
 LOG.info("dom.getCombustibleId(): "+dom.getTipoDatoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoDato)businessCatTipoDato.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDato.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoDato update(@RequestBody DominioCatTipoDato dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoDato.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoDato)businessCatTipoDato.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDato.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoDato delete(@RequestBody DominioCatTipoDato dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoDato.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoDato)businessCatTipoDato.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDato.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoDato findById(@RequestBody DominioCatTipoDato dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoDato.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoDato)businessCatTipoDato.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDato.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoDato count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoDato.count. . .");  
 DominioCatTipoDato dom = new DominioCatTipoDato();  
 dom.setCount(businessCatTipoDato.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDato.count: "+ex);  
 DominioCatTipoDato dom = new DominioCatTipoDato();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoDato existsById(@RequestBody DominioCatTipoDato dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoDato.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatTipoDato)businessCatTipoDato.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDato.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatTipoDocumento.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatTipoDocumento;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatTipoDocumento;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatTipoDocumento")  
public class RESTCatTipoDocumento {  
 private static final Logger LOG = LogManager.getLogger(RESTCatTipoDocumento.class);  
 @Autowired  
 private IfaceCatTipoDocumento businessCatTipoDocumento;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatTipoDocumento> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoDocumento.findAll. . .");  
 List<DominioCatTipoDocumento> lista = businessCatTipoDocumento.findAll();  
 LOG.info("invoke RESTCatTipoDocumento.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDocumento.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatTipoDocumento dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatTipoDocumento.findAllPagination. . .");  
 response = businessCatTipoDocumento.allPagination(dom);  
 LOG.info("invoke RESTCatTipoDocumento.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDocumento.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatTipoDocumento.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoDocumento save(@RequestBody DominioCatTipoDocumento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoDocumento.save. . .");  
 LOG.info("dom.getTipoDocumentoId(): "+dom.getTipoDocumentoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoDocumento)businessCatTipoDocumento.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDocumento.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoDocumento update(@RequestBody DominioCatTipoDocumento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoDocumento.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoDocumento)businessCatTipoDocumento.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDocumento.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoDocumento delete(@RequestBody DominioCatTipoDocumento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoDocumento.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoDocumento)businessCatTipoDocumento.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDocumento.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoDocumento findById(@RequestBody DominioCatTipoDocumento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoDocumento.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoDocumento)businessCatTipoDocumento.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDocumento.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoDocumento count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoDocumento.count. . .");  
 DominioCatTipoDocumento dom = new DominioCatTipoDocumento();  
 dom.setCount(businessCatTipoDocumento.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDocumento.count: "+ex);  
 DominioCatTipoDocumento dom = new DominioCatTipoDocumento();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoDocumento existsById(@RequestBody DominioCatTipoDocumento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoDocumento.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatTipoDocumento)businessCatTipoDocumento.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoDocumento.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatTipoFactoresSat.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatTipoFactoresSat;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatTipoFactoresSat;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatTipoFactoresSat")  
public class RESTCatTipoFactoresSat {  
 private static final Logger LOG = LogManager.getLogger(RESTCatTipoFactoresSat.class);  
 @Autowired  
 private IfaceCatTipoFactoresSat businessCatTipoFactoresSat;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatTipoFactoresSat> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoFactoresSat.findAll. . .");  
 List<DominioCatTipoFactoresSat> lista = businessCatTipoFactoresSat.findAll();  
 LOG.info("invoke RESTCatTipoFactoresSat.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoFactoresSat.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatTipoFactoresSat dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatTipoFactoresSat.findAllPagination. . .");  
 response = businessCatTipoFactoresSat.allPagination(dom);  
 LOG.info("invoke RESTCatTipoFactoresSat.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoFactoresSat.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatTipoFactoresSat.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoFactoresSat save(@RequestBody DominioCatTipoFactoresSat dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoFactoresSat.save. . .");  
 LOG.info("dom.getTipoFactorSatId(): "+dom.getTipoFactorSatId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoFactoresSat)businessCatTipoFactoresSat.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoFactoresSat.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoFactoresSat update(@RequestBody DominioCatTipoFactoresSat dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoFactoresSat.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoFactoresSat)businessCatTipoFactoresSat.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoFactoresSat.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoFactoresSat delete(@RequestBody DominioCatTipoFactoresSat dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoFactoresSat.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoFactoresSat)businessCatTipoFactoresSat.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoFactoresSat.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoFactoresSat findById(@RequestBody DominioCatTipoFactoresSat dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoFactoresSat.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoFactoresSat)businessCatTipoFactoresSat.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoFactoresSat.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoFactoresSat count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoFactoresSat.count. . .");  
 DominioCatTipoFactoresSat dom = new DominioCatTipoFactoresSat();  
 dom.setCount(businessCatTipoFactoresSat.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoFactoresSat.count: "+ex);  
 DominioCatTipoFactoresSat dom = new DominioCatTipoFactoresSat();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoFactoresSat existsById(@RequestBody DominioCatTipoFactoresSat dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoFactoresSat.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatTipoFactoresSat)businessCatTipoFactoresSat.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoFactoresSat.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatTipoImpuestos.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatTipoImpuestos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatTipoImpuestos;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatTipoImpuestos")  
public class RESTCatTipoImpuestos {  
 private static final Logger LOG = LogManager.getLogger(RESTCatTipoImpuestos.class);  
 @Autowired  
 private IfaceCatTipoImpuestos businessCatTipoImpuestos;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatTipoImpuestos> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoImpuestos.findAll. . .");  
 List<DominioCatTipoImpuestos> lista = businessCatTipoImpuestos.findAll();  
 LOG.info("invoke RESTCatTipoImpuestos.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoImpuestos.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatTipoImpuestos dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatTipoImpuestos.findAllPagination. . .");  
 response = businessCatTipoImpuestos.allPagination(dom);  
 LOG.info("invoke RESTCatTipoImpuestos.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoImpuestos.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatTipoImpuestos.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoImpuestos save(@RequestBody DominioCatTipoImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoImpuestos.save. . .");  
 LOG.info("dom.getTipoImpuestoId(): "+dom.getTipoImpuestoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoImpuestos)businessCatTipoImpuestos.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoImpuestos.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoImpuestos update(@RequestBody DominioCatTipoImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoImpuestos.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoImpuestos)businessCatTipoImpuestos.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoImpuestos.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoImpuestos delete(@RequestBody DominioCatTipoImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoImpuestos.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoImpuestos)businessCatTipoImpuestos.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoImpuestos.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoImpuestos findById(@RequestBody DominioCatTipoImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoImpuestos.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoImpuestos)businessCatTipoImpuestos.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoImpuestos.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoImpuestos count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoImpuestos.count. . .");  
 DominioCatTipoImpuestos dom = new DominioCatTipoImpuestos();  
 dom.setCount(businessCatTipoImpuestos.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoImpuestos.count: "+ex);  
 DominioCatTipoImpuestos dom = new DominioCatTipoImpuestos();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoImpuestos existsById(@RequestBody DominioCatTipoImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoImpuestos.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatTipoImpuestos)businessCatTipoImpuestos.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoImpuestos.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatTipoRelacion.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatTipoRelacion;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatTipoRelacion;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatTipoRelacion")  
public class RESTCatTipoRelacion {  
 private static final Logger LOG = LogManager.getLogger(RESTCatTipoRelacion.class);  
 @Autowired  
 private IfaceCatTipoRelacion businessCatTipoRelacion;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatTipoRelacion> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoRelacion.findAll. . .");  
 List<DominioCatTipoRelacion> lista = businessCatTipoRelacion.findAll();  
 LOG.info("invoke RESTCatTipoRelacion.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoRelacion.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatTipoRelacion dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatTipoRelacion.findAllPagination. . .");  
 response = businessCatTipoRelacion.allPagination(dom);  
 LOG.info("invoke RESTCatTipoRelacion.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoRelacion.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatTipoRelacion.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoRelacion save(@RequestBody DominioCatTipoRelacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoRelacion.save. . .");  
 LOG.info("dom.getTipoRelacionId(): "+dom.getTipoRelacionId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoRelacion)businessCatTipoRelacion.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoRelacion.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoRelacion update(@RequestBody DominioCatTipoRelacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoRelacion.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoRelacion)businessCatTipoRelacion.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoRelacion.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoRelacion delete(@RequestBody DominioCatTipoRelacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoRelacion.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoRelacion)businessCatTipoRelacion.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoRelacion.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoRelacion findById(@RequestBody DominioCatTipoRelacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoRelacion.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTipoRelacion)businessCatTipoRelacion.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoRelacion.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoRelacion count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoRelacion.count. . .");  
 DominioCatTipoRelacion dom = new DominioCatTipoRelacion();  
 dom.setCount(businessCatTipoRelacion.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoRelacion.count: "+ex);  
 DominioCatTipoRelacion dom = new DominioCatTipoRelacion();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTipoRelacion existsById(@RequestBody DominioCatTipoRelacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTipoRelacion.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatTipoRelacion)businessCatTipoRelacion.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTipoRelacion.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatTiposDeEvento.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatTiposDeEvento;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatTiposDeEvento;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatTiposDeEvento")  
public class RESTCatTiposDeEvento {  
 private static final Logger LOG = LogManager.getLogger(RESTCatTiposDeEvento.class);  
 @Autowired  
 private IfaceCatTiposDeEvento businessCatTiposDeEvento;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatTiposDeEvento> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposDeEvento.findAll. . .");  
 List<DominioCatTiposDeEvento> lista = businessCatTiposDeEvento.findAll();  
 LOG.info("invoke RESTCatTiposDeEvento.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposDeEvento.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatTiposDeEvento dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatTiposDeEvento.findAllPagination. . .");  
 response = businessCatTiposDeEvento.allPagination(dom);  
 LOG.info("invoke RESTCatTiposDeEvento.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposDeEvento.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatTiposDeEvento.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposDeEvento save(@RequestBody DominioCatTiposDeEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposDeEvento.save. . .");  
 LOG.info("dom.getTiposDeEventoId(): "+dom.getTiposDeEventoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTiposDeEvento)businessCatTiposDeEvento.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposDeEvento.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposDeEvento update(@RequestBody DominioCatTiposDeEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposDeEvento.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTiposDeEvento)businessCatTiposDeEvento.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposDeEvento.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposDeEvento delete(@RequestBody DominioCatTiposDeEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposDeEvento.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTiposDeEvento)businessCatTiposDeEvento.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposDeEvento.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposDeEvento findById(@RequestBody DominioCatTiposDeEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposDeEvento.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTiposDeEvento)businessCatTiposDeEvento.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposDeEvento.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposDeEvento count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposDeEvento.count. . .");  
 DominioCatTiposDeEvento dom = new DominioCatTiposDeEvento();  
 dom.setCount(businessCatTiposDeEvento.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposDeEvento.count: "+ex);  
 DominioCatTiposDeEvento dom = new DominioCatTiposDeEvento();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposDeEvento existsById(@RequestBody DominioCatTiposDeEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposDeEvento.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatTiposDeEvento)businessCatTiposDeEvento.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposDeEvento.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatTiposFacturas.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatTiposFacturas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatTiposFacturas;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatTiposFacturas")  
public class RESTCatTiposFacturas {  
 private static final Logger LOG = LogManager.getLogger(RESTCatTiposFacturas.class);  
 @Autowired  
 private IfaceCatTiposFacturas businessCatTiposFacturas;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatTiposFacturas> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposFacturas.findAll. . .");  
 List<DominioCatTiposFacturas> lista = businessCatTiposFacturas.findAll();  
 LOG.info("invoke RESTCatTiposFacturas.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposFacturas.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatTiposFacturas dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatTiposFacturas.findAllPagination. . .");  
 response = businessCatTiposFacturas.allPagination(dom);  
 LOG.info("invoke RESTCatTiposFacturas.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposFacturas.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatTiposFacturas.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposFacturas save(@RequestBody DominioCatTiposFacturas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposFacturas.save. . .");  
 LOG.info("dom.getTipoFacturaId(): "+dom.getTipoFacturaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTiposFacturas)businessCatTiposFacturas.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposFacturas.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposFacturas update(@RequestBody DominioCatTiposFacturas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposFacturas.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTiposFacturas)businessCatTiposFacturas.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposFacturas.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposFacturas delete(@RequestBody DominioCatTiposFacturas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposFacturas.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTiposFacturas)businessCatTiposFacturas.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposFacturas.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposFacturas findById(@RequestBody DominioCatTiposFacturas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposFacturas.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTiposFacturas)businessCatTiposFacturas.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposFacturas.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposFacturas count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposFacturas.count. . .");  
 DominioCatTiposFacturas dom = new DominioCatTiposFacturas();  
 dom.setCount(businessCatTiposFacturas.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposFacturas.count: "+ex);  
 DominioCatTiposFacturas dom = new DominioCatTiposFacturas();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposFacturas existsById(@RequestBody DominioCatTiposFacturas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposFacturas.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatTiposFacturas)businessCatTiposFacturas.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposFacturas.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatTiposServicios.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatTiposServicios;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatTiposServicios;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatTiposServicios")  
public class RESTCatTiposServicios {  
 private static final Logger LOG = LogManager.getLogger(RESTCatTiposServicios.class);  
 @Autowired  
 private IfaceCatTiposServicios businessCatTiposServicios;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatTiposServicios> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposServicios.findAll. . .");  
 List<DominioCatTiposServicios> lista = businessCatTiposServicios.findAll();  
 LOG.info("invoke RESTCatTiposServicios.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposServicios.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatTiposServicios dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatTiposServicios.findAllPagination. . .");  
 response = businessCatTiposServicios.allPagination(dom);  
 LOG.info("invoke RESTCatTiposServicios.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposServicios.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatTiposServicios.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposServicios save(@RequestBody DominioCatTiposServicios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposServicios.save. . .");  
 LOG.info("dom.getTipoServicioId(): "+dom.getTipoServicioId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTiposServicios)businessCatTiposServicios.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposServicios.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposServicios update(@RequestBody DominioCatTiposServicios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposServicios.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTiposServicios)businessCatTiposServicios.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposServicios.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposServicios delete(@RequestBody DominioCatTiposServicios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposServicios.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTiposServicios)businessCatTiposServicios.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposServicios.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposServicios findById(@RequestBody DominioCatTiposServicios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposServicios.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatTiposServicios)businessCatTiposServicios.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposServicios.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposServicios count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposServicios.count. . .");  
 DominioCatTiposServicios dom = new DominioCatTiposServicios();  
 dom.setCount(businessCatTiposServicios.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposServicios.count: "+ex);  
 DominioCatTiposServicios dom = new DominioCatTiposServicios();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatTiposServicios existsById(@RequestBody DominioCatTiposServicios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatTiposServicios.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatTiposServicios)businessCatTiposServicios.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatTiposServicios.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatUnidad.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatUnidad;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatUnidad;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatUnidad")  
public class RESTCatUnidad {  
 private static final Logger LOG = LogManager.getLogger(RESTCatUnidad.class);  
 @Autowired  
 private IfaceCatUnidad businessCatUnidad;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatUnidad> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatUnidad.findAll. . .");  
 List<DominioCatUnidad> lista = businessCatUnidad.findAll();  
 LOG.info("invoke RESTCatUnidad.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidad.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatUnidad dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatUnidad.findAllPagination. . .");  
 response = businessCatUnidad.allPagination(dom);  
 LOG.info("invoke RESTCatUnidad.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidad.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatUnidad.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUnidad save(@RequestBody DominioCatUnidad dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatUnidad.save. . .");  
 LOG.info("dom.getUnidadId(): "+dom.getUnidadId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatUnidad)businessCatUnidad.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidad.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUnidad update(@RequestBody DominioCatUnidad dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatUnidad.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatUnidad)businessCatUnidad.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidad.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUnidad delete(@RequestBody DominioCatUnidad dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatUnidad.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatUnidad)businessCatUnidad.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidad.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUnidad findById(@RequestBody DominioCatUnidad dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatUnidad.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatUnidad)businessCatUnidad.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidad.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUnidad count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatUnidad.count. . .");  
 DominioCatUnidad dom = new DominioCatUnidad();  
 dom.setCount(businessCatUnidad.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidad.count: "+ex);  
 DominioCatUnidad dom = new DominioCatUnidad();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUnidad existsById(@RequestBody DominioCatUnidad dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatUnidad.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatUnidad)businessCatUnidad.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidad.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatUnidades.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatUnidades;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatUnidades;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatUnidades")  
public class RESTCatUnidades {  
 private static final Logger LOG = LogManager.getLogger(RESTCatUnidades.class);  
 @Autowired  
 private IfaceCatUnidades businessCatUnidades;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatUnidades> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatUnidades.findAll. . .");  
 List<DominioCatUnidades> lista = businessCatUnidades.findAll();  
 LOG.info("invoke RESTCatUnidades.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidades.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatUnidades dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatUnidades.findAllPagination. . .");  
 response = businessCatUnidades.allPagination(dom);  
 LOG.info("invoke RESTCatUnidades.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidades.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatUnidades.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUnidades save(@RequestBody DominioCatUnidades dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatUnidades.save. . .");  
 LOG.info("dom.getUnidadId(): "+dom.getUnidadId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatUnidades)businessCatUnidades.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidades.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUnidades update(@RequestBody DominioCatUnidades dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatUnidades.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatUnidades)businessCatUnidades.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidades.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUnidades delete(@RequestBody DominioCatUnidades dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatUnidades.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatUnidades)businessCatUnidades.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidades.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUnidades findById(@RequestBody DominioCatUnidades dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatUnidades.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatUnidades)businessCatUnidades.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidades.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUnidades count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatUnidades.count. . .");  
 DominioCatUnidades dom = new DominioCatUnidades();  
 dom.setCount(businessCatUnidades.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidades.count: "+ex);  
 DominioCatUnidades dom = new DominioCatUnidades();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUnidades existsById(@RequestBody DominioCatUnidades dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatUnidades.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatUnidades)businessCatUnidades.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUnidades.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatUsoCfdi.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatUsoCfdi;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatUsoCfdi;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatUsoCfdi")  
public class RESTCatUsoCfdi {  
 private static final Logger LOG = LogManager.getLogger(RESTCatUsoCfdi.class);  
 @Autowired  
 private IfaceCatUsoCfdi businessCatUsoCfdi;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatUsoCfdi> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatUsoCfdi.findAll. . .");  
 List<DominioCatUsoCfdi> lista = businessCatUsoCfdi.findAll();  
 LOG.info("invoke RESTCatUsoCfdi.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUsoCfdi.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatUsoCfdi dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatUsoCfdi.findAllPagination. . .");  
 response = businessCatUsoCfdi.allPagination(dom);  
 LOG.info("invoke RESTCatUsoCfdi.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUsoCfdi.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatUsoCfdi.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUsoCfdi save(@RequestBody DominioCatUsoCfdi dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatUsoCfdi.save. . .");  
 LOG.info("dom.getUsoCfdiId(): "+dom.getUsoCfdiId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatUsoCfdi)businessCatUsoCfdi.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUsoCfdi.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUsoCfdi update(@RequestBody DominioCatUsoCfdi dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatUsoCfdi.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatUsoCfdi)businessCatUsoCfdi.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUsoCfdi.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUsoCfdi delete(@RequestBody DominioCatUsoCfdi dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatUsoCfdi.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatUsoCfdi)businessCatUsoCfdi.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUsoCfdi.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUsoCfdi findById(@RequestBody DominioCatUsoCfdi dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatUsoCfdi.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatUsoCfdi)businessCatUsoCfdi.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUsoCfdi.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUsoCfdi count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatUsoCfdi.count. . .");  
 DominioCatUsoCfdi dom = new DominioCatUsoCfdi();  
 dom.setCount(businessCatUsoCfdi.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUsoCfdi.count: "+ex);  
 DominioCatUsoCfdi dom = new DominioCatUsoCfdi();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatUsoCfdi existsById(@RequestBody DominioCatUsoCfdi dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatUsoCfdi.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatUsoCfdi)businessCatUsoCfdi.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatUsoCfdi.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatVariable.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatVariable;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatVariable;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatVariable")  
public class RESTCatVariable {  
 private static final Logger LOG = LogManager.getLogger(RESTCatVariable.class);  
 @Autowired  
 private IfaceCatVariable businessCatVariable;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatVariable> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatVariable.findAll. . .");  
 List<DominioCatVariable> lista = businessCatVariable.findAll();  
 LOG.info("invoke RESTCatVariable.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariable.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatVariable dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatVariable.findAllPagination. . .");  
 response = businessCatVariable.allPagination(dom);  
 LOG.info("invoke RESTCatVariable.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariable.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatVariable.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVariable save(@RequestBody DominioCatVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatVariable.save. . .");  
 LOG.info("dom.getVariableId(): "+dom.getVariableId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatVariable)businessCatVariable.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariable.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVariable update(@RequestBody DominioCatVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatVariable.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatVariable)businessCatVariable.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariable.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVariable delete(@RequestBody DominioCatVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatVariable.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatVariable)businessCatVariable.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariable.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVariable findById(@RequestBody DominioCatVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatVariable.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatVariable)businessCatVariable.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariable.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVariable count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatVariable.count. . .");  
 DominioCatVariable dom = new DominioCatVariable();  
 dom.setCount(businessCatVariable.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariable.count: "+ex);  
 DominioCatVariable dom = new DominioCatVariable();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVariable existsById(@RequestBody DominioCatVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatVariable.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatVariable)businessCatVariable.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariable.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatVariablePerfil.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatVariablePerfil;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatVariablePerfil;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatVariablePerfil")  
public class RESTCatVariablePerfil {  
 private static final Logger LOG = LogManager.getLogger(RESTCatVariablePerfil.class);  
 @Autowired  
 private IfaceCatVariablePerfil businessCatVariablePerfil;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatVariablePerfil> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatVariablePerfil.findAll. . .");  
 List<DominioCatVariablePerfil> lista = businessCatVariablePerfil.findAll();  
 LOG.info("invoke RESTCatVariablePerfil.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariablePerfil.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatVariablePerfil dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatVariablePerfil.findAllPagination. . .");  
 response = businessCatVariablePerfil.allPagination(dom);  
 LOG.info("invoke RESTCatVariablePerfil.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariablePerfil.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatVariablePerfil.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVariablePerfil save(@RequestBody DominioCatVariablePerfil dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatVariablePerfil.save. . .");  
 LOG.info("dom.getVariablePerfilId(): "+dom.getVariablePerfilId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatVariablePerfil)businessCatVariablePerfil.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariablePerfil.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVariablePerfil update(@RequestBody DominioCatVariablePerfil dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatVariablePerfil.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatVariablePerfil)businessCatVariablePerfil.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariablePerfil.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVariablePerfil delete(@RequestBody DominioCatVariablePerfil dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatVariablePerfil.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatVariablePerfil)businessCatVariablePerfil.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariablePerfil.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVariablePerfil findById(@RequestBody DominioCatVariablePerfil dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatVariablePerfil.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatVariablePerfil)businessCatVariablePerfil.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariablePerfil.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVariablePerfil count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatVariablePerfil.count. . .");  
 DominioCatVariablePerfil dom = new DominioCatVariablePerfil();  
 dom.setCount(businessCatVariablePerfil.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariablePerfil.count: "+ex);  
 DominioCatVariablePerfil dom = new DominioCatVariablePerfil();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVariablePerfil existsById(@RequestBody DominioCatVariablePerfil dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatVariablePerfil.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatVariablePerfil)businessCatVariablePerfil.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVariablePerfil.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTCatVersion.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceCatVersion;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioCatVersion;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/CatVersion")  
public class RESTCatVersion {  
 private static final Logger LOG = LogManager.getLogger(RESTCatVersion.class);  
 @Autowired  
 private IfaceCatVersion businessCatVersion;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioCatVersion> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTCatVersion.findAll. . .");  
 List<DominioCatVersion> lista = businessCatVersion.findAll();  
 LOG.info("invoke RESTCatVersion.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVersion.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioCatVersion dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTCatVersion.findAllPagination. . .");  
 response = businessCatVersion.allPagination(dom);  
 LOG.info("invoke RESTCatVersion.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVersion.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTCatVersion.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVersion save(@RequestBody DominioCatVersion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatVersion.save. . .");  
 LOG.info("dom.getVersionId(): "+dom.getVersionId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatVersion)businessCatVersion.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVersion.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVersion update(@RequestBody DominioCatVersion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatVersion.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatVersion)businessCatVersion.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVersion.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVersion delete(@RequestBody DominioCatVersion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatVersion.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatVersion)businessCatVersion.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVersion.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVersion findById(@RequestBody DominioCatVersion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatVersion.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioCatVersion)businessCatVersion.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVersion.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVersion count() throws Exception {  
 try {  
 LOG.info("invoke RESTCatVersion.count. . .");  
 DominioCatVersion dom = new DominioCatVersion();  
 dom.setCount(businessCatVersion.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVersion.count: "+ex);  
 DominioCatVersion dom = new DominioCatVersion();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioCatVersion existsById(@RequestBody DominioCatVersion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTCatVersion.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioCatVersion)businessCatVersion.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTCatVersion.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTSatImpuestos.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceSatImpuestos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatImpuestos;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/SatImpuestos")  
public class RESTSatImpuestos {  
 private static final Logger LOG = LogManager.getLogger(RESTSatImpuestos.class);  
 @Autowired  
 private IfaceSatImpuestos businessSatImpuestos;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioSatImpuestos> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTSatImpuestos.findAll. . .");  
 List<DominioSatImpuestos> lista = businessSatImpuestos.findAll();  
 LOG.info("invoke RESTSatImpuestos.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatImpuestos.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioSatImpuestos dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTSatImpuestos.findAllPagination. . .");  
 response = businessSatImpuestos.allPagination(dom);  
 LOG.info("invoke RESTSatImpuestos.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatImpuestos.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTSatImpuestos.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatImpuestos save(@RequestBody DominioSatImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatImpuestos.save. . .");  
 LOG.info("dom.getImpuestoId(): "+dom.getImpuestoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatImpuestos)businessSatImpuestos.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatImpuestos.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatImpuestos update(@RequestBody DominioSatImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatImpuestos.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatImpuestos)businessSatImpuestos.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatImpuestos.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatImpuestos delete(@RequestBody DominioSatImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatImpuestos.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatImpuestos)businessSatImpuestos.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatImpuestos.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatImpuestos findById(@RequestBody DominioSatImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatImpuestos.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatImpuestos)businessSatImpuestos.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatImpuestos.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatImpuestos count() throws Exception {  
 try {  
 LOG.info("invoke RESTSatImpuestos.count. . .");  
 DominioSatImpuestos dom = new DominioSatImpuestos();  
 dom.setCount(businessSatImpuestos.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatImpuestos.count: "+ex);  
 DominioSatImpuestos dom = new DominioSatImpuestos();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatImpuestos existsById(@RequestBody DominioSatImpuestos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatImpuestos.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioSatImpuestos)businessSatImpuestos.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatImpuestos.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTSatMonedas.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceSatMonedas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatMonedas;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/SatMonedas")  
public class RESTSatMonedas {  
 private static final Logger LOG = LogManager.getLogger(RESTSatMonedas.class);  
 @Autowired  
 private IfaceSatMonedas businessSatMonedas;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioSatMonedas> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTSatMonedas.findAll. . .");  
 List<DominioSatMonedas> lista = businessSatMonedas.findAll();  
 LOG.info("invoke RESTSatMonedas.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatMonedas.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioSatMonedas dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTSatMonedas.findAllPagination. . .");  
 response = businessSatMonedas.allPagination(dom);  
 LOG.info("invoke RESTSatMonedas.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatMonedas.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTSatMonedas.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatMonedas save(@RequestBody DominioSatMonedas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatMonedas.save. . .");  
 LOG.info("dom.getMonedaSatId(): "+dom.getMonedaSatId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatMonedas)businessSatMonedas.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatMonedas.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatMonedas update(@RequestBody DominioSatMonedas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatMonedas.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatMonedas)businessSatMonedas.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatMonedas.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatMonedas delete(@RequestBody DominioSatMonedas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatMonedas.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatMonedas)businessSatMonedas.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatMonedas.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatMonedas findById(@RequestBody DominioSatMonedas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatMonedas.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatMonedas)businessSatMonedas.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatMonedas.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatMonedas count() throws Exception {  
 try {  
 LOG.info("invoke RESTSatMonedas.count. . .");  
 DominioSatMonedas dom = new DominioSatMonedas();  
 dom.setCount(businessSatMonedas.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatMonedas.count: "+ex);  
 DominioSatMonedas dom = new DominioSatMonedas();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatMonedas existsById(@RequestBody DominioSatMonedas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatMonedas.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioSatMonedas)businessSatMonedas.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatMonedas.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTSatPaises.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceSatPaises;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatPaises;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/SatPaises")  
public class RESTSatPaises {  
 private static final Logger LOG = LogManager.getLogger(RESTSatPaises.class);  
 @Autowired  
 private IfaceSatPaises businessSatPaises;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioSatPaises> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTSatPaises.findAll. . .");  
 List<DominioSatPaises> lista = businessSatPaises.findAll();  
 LOG.info("invoke RESTSatPaises.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatPaises.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioSatPaises dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTSatPaises.findAllPagination. . .");  
 response = businessSatPaises.allPagination(dom);  
 LOG.info("invoke RESTSatPaises.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatPaises.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTSatPaises.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatPaises save(@RequestBody DominioSatPaises dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatPaises.save. . .");  
 LOG.info("dom.getPaisSatId(): "+dom.getPaisSatId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatPaises)businessSatPaises.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatPaises.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatPaises update(@RequestBody DominioSatPaises dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatPaises.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatPaises)businessSatPaises.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatPaises.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatPaises delete(@RequestBody DominioSatPaises dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatPaises.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatPaises)businessSatPaises.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatPaises.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatPaises findById(@RequestBody DominioSatPaises dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatPaises.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatPaises)businessSatPaises.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatPaises.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatPaises count() throws Exception {  
 try {  
 LOG.info("invoke RESTSatPaises.count. . .");  
 DominioSatPaises dom = new DominioSatPaises();  
 dom.setCount(businessSatPaises.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatPaises.count: "+ex);  
 DominioSatPaises dom = new DominioSatPaises();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatPaises existsById(@RequestBody DominioSatPaises dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatPaises.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioSatPaises)businessSatPaises.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatPaises.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTSatProductosClaves.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceSatProductosClaves;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatProductosClaves;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/SatProductosClaves")  
public class RESTSatProductosClaves {  
 private static final Logger LOG = LogManager.getLogger(RESTSatProductosClaves.class);  
 @Autowired  
 private IfaceSatProductosClaves businessSatProductosClaves;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioSatProductosClaves> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTSatProductosClaves.findAll. . .");  
 List<DominioSatProductosClaves> lista = businessSatProductosClaves.findAll();  
 LOG.info("invoke RESTSatProductosClaves.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosClaves.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioSatProductosClaves dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTSatProductosClaves.findAllPagination. . .");  
 response = businessSatProductosClaves.allPagination(dom);  
 LOG.info("invoke RESTSatProductosClaves.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosClaves.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTSatProductosClaves.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatProductosClaves save(@RequestBody DominioSatProductosClaves dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatProductosClaves.save. . .");  
 LOG.info("dom.getSatProductosClavesId(): "+dom.getSatProductosClavesId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatProductosClaves)businessSatProductosClaves.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosClaves.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatProductosClaves update(@RequestBody DominioSatProductosClaves dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatProductosClaves.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatProductosClaves)businessSatProductosClaves.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosClaves.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatProductosClaves delete(@RequestBody DominioSatProductosClaves dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatProductosClaves.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatProductosClaves)businessSatProductosClaves.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosClaves.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatProductosClaves findById(@RequestBody DominioSatProductosClaves dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatProductosClaves.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatProductosClaves)businessSatProductosClaves.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosClaves.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatProductosClaves count() throws Exception {  
 try {  
 LOG.info("invoke RESTSatProductosClaves.count. . .");  
 DominioSatProductosClaves dom = new DominioSatProductosClaves();  
 dom.setCount(businessSatProductosClaves.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosClaves.count: "+ex);  
 DominioSatProductosClaves dom = new DominioSatProductosClaves();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatProductosClaves existsById(@RequestBody DominioSatProductosClaves dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatProductosClaves.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioSatProductosClaves)businessSatProductosClaves.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosClaves.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTSatProductosUnidades.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceSatProductosUnidades;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatProductosUnidades;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/SatProductosUnidades")  
public class RESTSatProductosUnidades {  
 private static final Logger LOG = LogManager.getLogger(RESTSatProductosUnidades.class);  
 @Autowired  
 private IfaceSatProductosUnidades businessSatProductosUnidades;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioSatProductosUnidades> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTSatProductosUnidades.findAll. . .");  
 List<DominioSatProductosUnidades> lista = businessSatProductosUnidades.findAll();  
 LOG.info("invoke RESTSatProductosUnidades.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosUnidades.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioSatProductosUnidades dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTSatProductosUnidades.findAllPagination. . .");  
 response = businessSatProductosUnidades.allPagination(dom);  
 LOG.info("invoke RESTSatProductosUnidades.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosUnidades.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTSatProductosUnidades.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatProductosUnidades save(@RequestBody DominioSatProductosUnidades dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatProductosUnidades.save. . .");  
 LOG.info("dom.getSatProductosUnidadesId(): "+dom.getSatProductosUnidadesId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatProductosUnidades)businessSatProductosUnidades.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosUnidades.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatProductosUnidades update(@RequestBody DominioSatProductosUnidades dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatProductosUnidades.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatProductosUnidades)businessSatProductosUnidades.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosUnidades.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatProductosUnidades delete(@RequestBody DominioSatProductosUnidades dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatProductosUnidades.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatProductosUnidades)businessSatProductosUnidades.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosUnidades.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatProductosUnidades findById(@RequestBody DominioSatProductosUnidades dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatProductosUnidades.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatProductosUnidades)businessSatProductosUnidades.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosUnidades.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatProductosUnidades count() throws Exception {  
 try {  
 LOG.info("invoke RESTSatProductosUnidades.count. . .");  
 DominioSatProductosUnidades dom = new DominioSatProductosUnidades();  
 dom.setCount(businessSatProductosUnidades.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosUnidades.count: "+ex);  
 DominioSatProductosUnidades dom = new DominioSatProductosUnidades();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatProductosUnidades existsById(@RequestBody DominioSatProductosUnidades dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatProductosUnidades.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioSatProductosUnidades)businessSatProductosUnidades.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatProductosUnidades.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTSatTasaCuota.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceSatTasaCuota;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatTasaCuota;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/SatTasaCuota")  
public class RESTSatTasaCuota {  
 private static final Logger LOG = LogManager.getLogger(RESTSatTasaCuota.class);  
 @Autowired  
 private IfaceSatTasaCuota businessSatTasaCuota;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioSatTasaCuota> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTSatTasaCuota.findAll. . .");  
 List<DominioSatTasaCuota> lista = businessSatTasaCuota.findAll();  
 LOG.info("invoke RESTSatTasaCuota.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTasaCuota.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioSatTasaCuota dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTSatTasaCuota.findAllPagination. . .");  
 response = businessSatTasaCuota.allPagination(dom);  
 LOG.info("invoke RESTSatTasaCuota.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTasaCuota.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTSatTasaCuota.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTasaCuota save(@RequestBody DominioSatTasaCuota dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatTasaCuota.save. . .");  
 LOG.info("dom.getTasaCuotaId(): "+dom.getTasaCuotaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatTasaCuota)businessSatTasaCuota.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTasaCuota.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTasaCuota update(@RequestBody DominioSatTasaCuota dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatTasaCuota.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatTasaCuota)businessSatTasaCuota.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTasaCuota.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTasaCuota delete(@RequestBody DominioSatTasaCuota dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatTasaCuota.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatTasaCuota)businessSatTasaCuota.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTasaCuota.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTasaCuota findById(@RequestBody DominioSatTasaCuota dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatTasaCuota.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatTasaCuota)businessSatTasaCuota.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTasaCuota.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTasaCuota count() throws Exception {  
 try {  
 LOG.info("invoke RESTSatTasaCuota.count. . .");  
 DominioSatTasaCuota dom = new DominioSatTasaCuota();  
 dom.setCount(businessSatTasaCuota.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTasaCuota.count: "+ex);  
 DominioSatTasaCuota dom = new DominioSatTasaCuota();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTasaCuota existsById(@RequestBody DominioSatTasaCuota dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatTasaCuota.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioSatTasaCuota)businessSatTasaCuota.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTasaCuota.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTSatTipoFactor.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceSatTipoFactor;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatTipoFactor;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/SatTipoFactor")  
public class RESTSatTipoFactor {  
 private static final Logger LOG = LogManager.getLogger(RESTSatTipoFactor.class);  
 @Autowired  
 private IfaceSatTipoFactor businessSatTipoFactor;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioSatTipoFactor> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTSatTipoFactor.findAll. . .");  
 List<DominioSatTipoFactor> lista = businessSatTipoFactor.findAll();  
 LOG.info("invoke RESTSatTipoFactor.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoFactor.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioSatTipoFactor dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTSatTipoFactor.findAllPagination. . .");  
 response = businessSatTipoFactor.allPagination(dom);  
 LOG.info("invoke RESTSatTipoFactor.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoFactor.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTSatTipoFactor.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTipoFactor save(@RequestBody DominioSatTipoFactor dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatTipoFactor.save. . .");  
 LOG.info("dom.getTipoFactorId(): "+dom.getTipoFactorId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatTipoFactor)businessSatTipoFactor.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoFactor.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTipoFactor update(@RequestBody DominioSatTipoFactor dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatTipoFactor.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatTipoFactor)businessSatTipoFactor.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoFactor.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTipoFactor delete(@RequestBody DominioSatTipoFactor dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatTipoFactor.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatTipoFactor)businessSatTipoFactor.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoFactor.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTipoFactor findById(@RequestBody DominioSatTipoFactor dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatTipoFactor.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatTipoFactor)businessSatTipoFactor.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoFactor.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTipoFactor count() throws Exception {  
 try {  
 LOG.info("invoke RESTSatTipoFactor.count. . .");  
 DominioSatTipoFactor dom = new DominioSatTipoFactor();  
 dom.setCount(businessSatTipoFactor.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoFactor.count: "+ex);  
 DominioSatTipoFactor dom = new DominioSatTipoFactor();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTipoFactor existsById(@RequestBody DominioSatTipoFactor dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatTipoFactor.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioSatTipoFactor)businessSatTipoFactor.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoFactor.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTSatTipoRelacion.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceSatTipoRelacion;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatTipoRelacion;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/SatTipoRelacion")  
public class RESTSatTipoRelacion {  
 private static final Logger LOG = LogManager.getLogger(RESTSatTipoRelacion.class);  
 @Autowired  
 private IfaceSatTipoRelacion businessSatTipoRelacion;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioSatTipoRelacion> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTSatTipoRelacion.findAll. . .");  
 List<DominioSatTipoRelacion> lista = businessSatTipoRelacion.findAll();  
 LOG.info("invoke RESTSatTipoRelacion.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoRelacion.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioSatTipoRelacion dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTSatTipoRelacion.findAllPagination. . .");  
 response = businessSatTipoRelacion.allPagination(dom);  
 LOG.info("invoke RESTSatTipoRelacion.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoRelacion.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTSatTipoRelacion.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTipoRelacion save(@RequestBody DominioSatTipoRelacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatTipoRelacion.save. . .");  
 LOG.info("dom.getNombre(): "+dom.getNombre());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatTipoRelacion)businessSatTipoRelacion.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoRelacion.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTipoRelacion update(@RequestBody DominioSatTipoRelacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatTipoRelacion.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatTipoRelacion)businessSatTipoRelacion.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoRelacion.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTipoRelacion delete(@RequestBody DominioSatTipoRelacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatTipoRelacion.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatTipoRelacion)businessSatTipoRelacion.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoRelacion.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTipoRelacion findById(@RequestBody DominioSatTipoRelacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatTipoRelacion.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatTipoRelacion)businessSatTipoRelacion.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoRelacion.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTipoRelacion count() throws Exception {  
 try {  
 LOG.info("invoke RESTSatTipoRelacion.count. . .");  
 DominioSatTipoRelacion dom = new DominioSatTipoRelacion();  
 dom.setCount(businessSatTipoRelacion.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoRelacion.count: "+ex);  
 DominioSatTipoRelacion dom = new DominioSatTipoRelacion();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatTipoRelacion existsById(@RequestBody DominioSatTipoRelacion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatTipoRelacion.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioSatTipoRelacion)businessSatTipoRelacion.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatTipoRelacion.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTSatUsoCfdi.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceSatUsoCfdi;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioSatUsoCfdi;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/SatUsoCfdi")  
public class RESTSatUsoCfdi {  
 private static final Logger LOG = LogManager.getLogger(RESTSatUsoCfdi.class);  
 @Autowired  
 private IfaceSatUsoCfdi businessSatUsoCfdi;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioSatUsoCfdi> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTSatUsoCfdi.findAll. . .");  
 List<DominioSatUsoCfdi> lista = businessSatUsoCfdi.findAll();  
 LOG.info("invoke RESTSatUsoCfdi.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatUsoCfdi.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioSatUsoCfdi dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTSatUsoCfdi.findAllPagination. . .");  
 response = businessSatUsoCfdi.allPagination(dom);  
 LOG.info("invoke RESTSatUsoCfdi.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatUsoCfdi.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTSatUsoCfdi.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatUsoCfdi save(@RequestBody DominioSatUsoCfdi dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatUsoCfdi.save. . .");  
 LOG.info("dom.getDescripcion(): "+dom.getDescripcion());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatUsoCfdi)businessSatUsoCfdi.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatUsoCfdi.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatUsoCfdi update(@RequestBody DominioSatUsoCfdi dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatUsoCfdi.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatUsoCfdi)businessSatUsoCfdi.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatUsoCfdi.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatUsoCfdi delete(@RequestBody DominioSatUsoCfdi dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatUsoCfdi.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatUsoCfdi)businessSatUsoCfdi.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatUsoCfdi.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatUsoCfdi findById(@RequestBody DominioSatUsoCfdi dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatUsoCfdi.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioSatUsoCfdi)businessSatUsoCfdi.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatUsoCfdi.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatUsoCfdi count() throws Exception {  
 try {  
 LOG.info("invoke RESTSatUsoCfdi.count. . .");  
 DominioSatUsoCfdi dom = new DominioSatUsoCfdi();  
 dom.setCount(businessSatUsoCfdi.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatUsoCfdi.count: "+ex);  
 DominioSatUsoCfdi dom = new DominioSatUsoCfdi();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioSatUsoCfdi existsById(@RequestBody DominioSatUsoCfdi dom) throws Exception {  
 try {  
 LOG.info("invoke RESTSatUsoCfdi.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioSatUsoCfdi)businessSatUsoCfdi.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTSatUsoCfdi.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxAdjuntos.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceTxAdjuntos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxAdjuntos;  
import mx.mbi.dgc.dominio.DominioTxEvento;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxAdjuntos")  
public class RESTTxAdjuntos {  
 private static final Logger LOG = LogManager.getLogger(RESTTxAdjuntos.class);  
 @Autowired  
 private IfaceTxAdjuntos businessTxAdjuntos;  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAll", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioTxAdjuntos> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxAdjuntos.findAll. . .");  
 List<DominioTxAdjuntos> lista = businessTxAdjuntos.findAll();  
 LOG.info("invoke RESTTxAdjuntos.findAll size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxAdjuntos.findAll: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findByEventoId", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioTxAdjuntos> findByEventoId(@RequestBody DominioTxEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxAdjuntos.findByEventoId. . .");  
 List<DominioTxAdjuntos> lista = businessTxAdjuntos.findByEventoId(dom);  
 LOG.info("invoke RESTTxAdjuntos.findAll size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAllPagination", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioRespuesta findAllPagination(@RequestBody DominioTxAdjuntos dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxAdjuntos.findAllPagination. . .");  
 response = businessTxAdjuntos.allPagination(dom);  
 LOG.info("invoke RESTTxAdjuntos.findAllPagination size lista return: " + response.getData().size());  
 return response;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxAdjuntos.findAllPagination: " + ex);  
 response.setMensaje("Error in invoke RESTTxAdjuntos.findAllPagination: " + ex);  
 return response;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/save", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxAdjuntos save(@RequestBody DominioTxAdjuntos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxAdjuntos.save. . .");  
 LOG.info("dom.getAdjuntoId(): " + dom.getAdjuntoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxAdjuntos) businessTxAdjuntos.save(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxAdjuntos.save: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/update", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxAdjuntos update(@RequestBody DominioTxAdjuntos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxAdjuntos.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxAdjuntos) businessTxAdjuntos.update(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxAdjuntos.update: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/delete", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxAdjuntos delete(@RequestBody DominioTxAdjuntos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxAdjuntos.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxAdjuntos) businessTxAdjuntos.delete(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxAdjuntos.delete: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxAdjuntos findById(@RequestBody DominioTxAdjuntos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxAdjuntos.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxAdjuntos) businessTxAdjuntos.findById(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxAdjuntos.findById: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/count", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxAdjuntos count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxAdjuntos.count. . .");  
 DominioTxAdjuntos dom = new DominioTxAdjuntos();  
 dom.setCount(businessTxAdjuntos.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxAdjuntos.count: " + ex);  
 DominioTxAdjuntos dom = new DominioTxAdjuntos();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/existsById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxAdjuntos existsById(@RequestBody DominioTxAdjuntos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxAdjuntos.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxAdjuntos) businessTxAdjuntos.existsById(dom);  
 if (dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxAdjuntos.existsById: " + ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxBitacora.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxBitacora;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxBitacora;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxBitacora")  
public class RESTTxBitacora {  
 private static final Logger LOG = LogManager.getLogger(RESTTxBitacora.class);  
 @Autowired  
 private IfaceTxBitacora businessTxBitacora;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxBitacora> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxBitacora.findAll. . .");  
 List<DominioTxBitacora> lista = businessTxBitacora.findAll();  
 LOG.info("invoke RESTTxBitacora.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxBitacora.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxBitacora dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxBitacora.findAllPagination. . .");  
 response = businessTxBitacora.allPagination(dom);  
 LOG.info("invoke RESTTxBitacora.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxBitacora.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxBitacora.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxBitacora save(@RequestBody DominioTxBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxBitacora.save. . .");  
 LOG.info("dom.getBitacoraId(): "+dom.getBitacoraId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxBitacora)businessTxBitacora.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxBitacora.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxBitacora update(@RequestBody DominioTxBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxBitacora.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxBitacora)businessTxBitacora.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxBitacora.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxBitacora delete(@RequestBody DominioTxBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxBitacora.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxBitacora)businessTxBitacora.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxBitacora.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxBitacora findById(@RequestBody DominioTxBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxBitacora.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxBitacora)businessTxBitacora.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxBitacora.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxBitacora count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxBitacora.count. . .");  
 DominioTxBitacora dom = new DominioTxBitacora();  
 dom.setCount(businessTxBitacora.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxBitacora.count: "+ex);  
 DominioTxBitacora dom = new DominioTxBitacora();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxBitacora existsById(@RequestBody DominioTxBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxBitacora.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxBitacora)businessTxBitacora.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxBitacora.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxCincominutal.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxCincominutal;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxCincominutal;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxCincominutal")  
public class RESTTxCincominutal {  
 private static final Logger LOG = LogManager.getLogger(RESTTxCincominutal.class);  
 @Autowired  
 private IfaceTxCincominutal businessTxCincominutal;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxCincominutal> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxCincominutal.findAll. . .");  
 List<DominioTxCincominutal> lista = businessTxCincominutal.findAll();  
 LOG.info("invoke RESTTxCincominutal.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutal.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxCincominutal dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxCincominutal.findAllPagination. . .");  
 response = businessTxCincominutal.allPagination(dom);  
 LOG.info("invoke RESTTxCincominutal.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutal.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxCincominutal.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCincominutal save(@RequestBody DominioTxCincominutal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCincominutal.save. . .");  
 LOG.info("dom.getCincominutalId(): "+dom.getCincominutalId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCincominutal)businessTxCincominutal.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutal.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCincominutal update(@RequestBody DominioTxCincominutal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCincominutal.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCincominutal)businessTxCincominutal.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutal.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCincominutal delete(@RequestBody DominioTxCincominutal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCincominutal.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCincominutal)businessTxCincominutal.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutal.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCincominutal findById(@RequestBody DominioTxCincominutal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCincominutal.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCincominutal)businessTxCincominutal.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutal.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCincominutal count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxCincominutal.count. . .");  
 DominioTxCincominutal dom = new DominioTxCincominutal();  
 dom.setCount(businessTxCincominutal.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutal.count: "+ex);  
 DominioTxCincominutal dom = new DominioTxCincominutal();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCincominutal existsById(@RequestBody DominioTxCincominutal dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCincominutal.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxCincominutal)businessTxCincominutal.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutal.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxCincominutalVar.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceTxCincominutalVar;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxCincominutalVar;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
import mx.mbi.dgc.utils.JRUtil;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxCincominutalVar")  
public class RESTTxCincominutalVar {  
 private static final Logger LOG = LogManager.getLogger(RESTTxCincominutalVar.class);  
 @Autowired  
 private IfaceTxCincominutalVar businessTxCincominutalVar;  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAll", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioTxCincominutalVar> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxCincominutalVar.findAll. . .");  
 List<DominioTxCincominutalVar> lista = businessTxCincominutalVar.findAll();  
 LOG.info("invoke RESTTxCincominutalVar.findAll size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutalVar.findAll: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findByEstatusCnNotAndFcPrimera",  
 consumes = "application/json", produces = "application/json")  
 public @ResponseBody  
 List<DominioTxCincominutalVar> findByEstatusCnAndFcPrimera(@RequestBody  
 DominioTxCincominutalVar dom) throws Exception {  
 try {  
 System.out.println("invoke RESTTxCincominutalVar.findByEstatusCnNotAndFcPrimera. . .: " +  
 dom.getEstatusCn() + ", " + dom.getFcPrimera());  
 CatPlanta planta = new CatPlanta();  
 JRUtil.copyProperties(planta, dom.getTxCincominutal().getCatPlanta());  
 List<DominioTxCincominutalVar> lista =  
 businessTxCincominutalVar.findByEstatusCnNotAndFcPrimera(dom.getEstatusCn(), planta);  
 LOG.info("invoke RESTTxCincominutalVar.findByEstatusCnNotAndFcPrimera size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutalVar.findByEstatusCnNotAndFcPrimera: " + ex);  
 ex.printStackTrace();  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAllPagination", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioRespuesta findAllPagination(@RequestBody DominioTxCincominutalVar dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxCincominutalVar.findAllPagination. . .");  
 response = businessTxCincominutalVar.allPagination(dom);  
 LOG.info("invoke RESTTxCincominutalVar.findAllPagination size lista return: " + response.getData().size());  
 return response;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutalVar.findAllPagination: " + ex);  
 response.setMensaje("Error in invoke RESTTxCincominutalVar.findAllPagination: " + ex);  
 return response;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/save", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxCincominutalVar save(@RequestBody DominioTxCincominutalVar dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCincominutalVar.save. . .");  
 LOG.info("dom.getCincominutalVarId(): " + dom.getCincominutalVarId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCincominutalVar) businessTxCincominutalVar.save(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutalVar.save: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/update", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxCincominutalVar update(@RequestBody DominioTxCincominutalVar dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCincominutalVar.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCincominutalVar) businessTxCincominutalVar.update(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutalVar.update: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/delete", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxCincominutalVar delete(@RequestBody DominioTxCincominutalVar dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCincominutalVar.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCincominutalVar) businessTxCincominutalVar.delete(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutalVar.delete: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxCincominutalVar findById(@RequestBody DominioTxCincominutalVar dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCincominutalVar.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCincominutalVar) businessTxCincominutalVar.findById(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutalVar.findById: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/count", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxCincominutalVar count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxCincominutalVar.count. . .");  
 DominioTxCincominutalVar dom = new DominioTxCincominutalVar();  
 dom.setCount(businessTxCincominutalVar.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutalVar.count: " + ex);  
 DominioTxCincominutalVar dom = new DominioTxCincominutalVar();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/existsById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxCincominutalVar existsById(@RequestBody DominioTxCincominutalVar dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCincominutalVar.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxCincominutalVar) businessTxCincominutalVar.existsById(dom);  
 if (dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxCincominutalVar.existsById: " + ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxCrudos.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxCrudos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxCrudos;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxCrudos")  
public class RESTTxCrudos {  
 private static final Logger LOG = LogManager.getLogger(RESTTxCrudos.class);  
 @Autowired  
 private IfaceTxCrudos businessTxCrudos;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxCrudos> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudos.findAll. . .");  
 List<DominioTxCrudos> lista = businessTxCrudos.findAll();  
 LOG.info("invoke RESTTxCrudos.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudos.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxCrudos dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxCrudos.findAllPagination. . .");  
 response = businessTxCrudos.allPagination(dom);  
 LOG.info("invoke RESTTxCrudos.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudos.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxCrudos.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudos save(@RequestBody DominioTxCrudos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudos.save. . .");  
 LOG.info("dom.getCrudosId(): "+dom.getCrudosId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCrudos)businessTxCrudos.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudos.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudos update(@RequestBody DominioTxCrudos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudos.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCrudos)businessTxCrudos.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudos.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudos delete(@RequestBody DominioTxCrudos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudos.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCrudos)businessTxCrudos.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudos.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudos findById(@RequestBody DominioTxCrudos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudos.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCrudos)businessTxCrudos.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudos.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudos count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudos.count. . .");  
 DominioTxCrudos dom = new DominioTxCrudos();  
 dom.setCount(businessTxCrudos.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudos.count: "+ex);  
 DominioTxCrudos dom = new DominioTxCrudos();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudos existsById(@RequestBody DominioTxCrudos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudos.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxCrudos)businessTxCrudos.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudos.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxCrudosLimpios.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxCrudosLimpios;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxCrudosLimpios;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxCrudosLimpios")  
public class RESTTxCrudosLimpios {  
 private static final Logger LOG = LogManager.getLogger(RESTTxCrudosLimpios.class);  
 @Autowired  
 private IfaceTxCrudosLimpios businessTxCrudosLimpios;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxCrudosLimpios> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudosLimpios.findAll. . .");  
 List<DominioTxCrudosLimpios> lista = businessTxCrudosLimpios.findAll();  
 LOG.info("invoke RESTTxCrudosLimpios.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosLimpios.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxCrudosLimpios dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxCrudosLimpios.findAllPagination. . .");  
 response = businessTxCrudosLimpios.allPagination(dom);  
 LOG.info("invoke RESTTxCrudosLimpios.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosLimpios.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxCrudosLimpios.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudosLimpios save(@RequestBody DominioTxCrudosLimpios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudosLimpios.save. . .");  
 LOG.info("dom.getFirma(): "+dom.getFirma());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCrudosLimpios)businessTxCrudosLimpios.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosLimpios.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudosLimpios update(@RequestBody DominioTxCrudosLimpios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudosLimpios.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCrudosLimpios)businessTxCrudosLimpios.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosLimpios.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudosLimpios delete(@RequestBody DominioTxCrudosLimpios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudosLimpios.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCrudosLimpios)businessTxCrudosLimpios.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosLimpios.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudosLimpios findById(@RequestBody DominioTxCrudosLimpios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudosLimpios.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCrudosLimpios)businessTxCrudosLimpios.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosLimpios.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudosLimpios count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudosLimpios.count. . .");  
 DominioTxCrudosLimpios dom = new DominioTxCrudosLimpios();  
 dom.setCount(businessTxCrudosLimpios.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosLimpios.count: "+ex);  
 DominioTxCrudosLimpios dom = new DominioTxCrudosLimpios();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudosLimpios existsById(@RequestBody DominioTxCrudosLimpios dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudosLimpios.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxCrudosLimpios)businessTxCrudosLimpios.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosLimpios.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxCrudosSol.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxCrudosSol;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxCrudosSol;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxCrudosSol")  
public class RESTTxCrudosSol {  
 private static final Logger LOG = LogManager.getLogger(RESTTxCrudosSol.class);  
 @Autowired  
 private IfaceTxCrudosSol businessTxCrudosSol;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxCrudosSol> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudosSol.findAll. . .");  
 List<DominioTxCrudosSol> lista = businessTxCrudosSol.findAll();  
 LOG.info("invoke RESTTxCrudosSol.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosSol.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxCrudosSol dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxCrudosSol.findAllPagination. . .");  
 response = businessTxCrudosSol.allPagination(dom);  
 LOG.info("invoke RESTTxCrudosSol.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosSol.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxCrudosSol.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudosSol save(@RequestBody DominioTxCrudosSol dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudosSol.save. . .");  
 LOG.info("dom.getCrudosId(): "+dom.getCrudosId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCrudosSol)businessTxCrudosSol.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosSol.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudosSol update(@RequestBody DominioTxCrudosSol dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudosSol.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCrudosSol)businessTxCrudosSol.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosSol.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudosSol delete(@RequestBody DominioTxCrudosSol dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudosSol.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCrudosSol)businessTxCrudosSol.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosSol.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudosSol findById(@RequestBody DominioTxCrudosSol dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudosSol.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxCrudosSol)businessTxCrudosSol.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosSol.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudosSol count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudosSol.count. . .");  
 DominioTxCrudosSol dom = new DominioTxCrudosSol();  
 dom.setCount(businessTxCrudosSol.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosSol.count: "+ex);  
 DominioTxCrudosSol dom = new DominioTxCrudosSol();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxCrudosSol existsById(@RequestBody DominioTxCrudosSol dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxCrudosSol.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxCrudosSol)businessTxCrudosSol.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxCrudosSol.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxDocumentos.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceTxDocumentos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxDocumentos;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxDocumentos")  
public class RESTTxDocumentos {  
 private static final Logger LOG = LogManager.getLogger(RESTTxDocumentos.class);  
 @Autowired  
 private IfaceTxDocumentos businessTxDocumentos;  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAll", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioTxDocumentos> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxDocumentos.findAll. . .");  
 List<DominioTxDocumentos> lista = businessTxDocumentos.findAll();  
 LOG.info("invoke RESTTxDocumentos.findAll size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxDocumentos.findAll: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAllPagination", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioRespuesta findAllPagination(@RequestBody DominioTxDocumentos dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxDocumentos.findAllPagination. . .");  
 response = businessTxDocumentos.allPagination(dom);  
 LOG.info("invoke RESTTxDocumentos.findAllPagination size lista return: " + response.getData().size());  
 return response;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxDocumentos.findAllPagination: " + ex);  
 response.setMensaje("Error in invoke RESTTxDocumentos.findAllPagination: " + ex);  
 return response;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/save", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxDocumentos save(@RequestBody DominioTxDocumentos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxDocumentos.save. . .");  
 LOG.info("dom.getDocumentoId(): " + dom.getDocumentoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxDocumentos) businessTxDocumentos.save(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxDocumentos.save: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/update", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxDocumentos update(@RequestBody DominioTxDocumentos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxDocumentos.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxDocumentos) businessTxDocumentos.update(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxDocumentos.update: " + ex);  
 ex.printStackTrace();  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/delete", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxDocumentos delete(@RequestBody DominioTxDocumentos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxDocumentos.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxDocumentos) businessTxDocumentos.delete(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxDocumentos.delete: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxDocumentos findById(@RequestBody DominioTxDocumentos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxDocumentos.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxDocumentos) businessTxDocumentos.findById(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxDocumentos.findById: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/count", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxDocumentos count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxDocumentos.count. . .");  
 DominioTxDocumentos dom = new DominioTxDocumentos();  
 dom.setCount(businessTxDocumentos.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxDocumentos.count: " + ex);  
 DominioTxDocumentos dom = new DominioTxDocumentos();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/existsById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxDocumentos existsById(@RequestBody DominioTxDocumentos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxDocumentos.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxDocumentos) businessTxDocumentos.existsById(dom);  
 if (dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxDocumentos.existsById: " + ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxErrorCargaCrudos.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxErrorCargaCrudos;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxErrorCargaCrudos;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxErrorCargaCrudos")  
public class RESTTxErrorCargaCrudos {  
 private static final Logger LOG = LogManager.getLogger(RESTTxErrorCargaCrudos.class);  
 @Autowired  
 private IfaceTxErrorCargaCrudos businessTxErrorCargaCrudos;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxErrorCargaCrudos> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxErrorCargaCrudos.findAll. . .");  
 List<DominioTxErrorCargaCrudos> lista = businessTxErrorCargaCrudos.findAll();  
 LOG.info("invoke RESTTxErrorCargaCrudos.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorCargaCrudos.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxErrorCargaCrudos dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxErrorCargaCrudos.findAllPagination. . .");  
 response = businessTxErrorCargaCrudos.allPagination(dom);  
 LOG.info("invoke RESTTxErrorCargaCrudos.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorCargaCrudos.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxErrorCargaCrudos.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxErrorCargaCrudos save(@RequestBody DominioTxErrorCargaCrudos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxErrorCargaCrudos.save. . .");  
 LOG.info("dom.getErrorCargaCrudosId(): "+dom.getErrorCargaCrudosId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxErrorCargaCrudos)businessTxErrorCargaCrudos.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorCargaCrudos.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxErrorCargaCrudos update(@RequestBody DominioTxErrorCargaCrudos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxErrorCargaCrudos.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxErrorCargaCrudos)businessTxErrorCargaCrudos.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorCargaCrudos.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxErrorCargaCrudos delete(@RequestBody DominioTxErrorCargaCrudos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxErrorCargaCrudos.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxErrorCargaCrudos)businessTxErrorCargaCrudos.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorCargaCrudos.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxErrorCargaCrudos findById(@RequestBody DominioTxErrorCargaCrudos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxErrorCargaCrudos.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxErrorCargaCrudos)businessTxErrorCargaCrudos.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorCargaCrudos.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxErrorCargaCrudos count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxErrorCargaCrudos.count. . .");  
 DominioTxErrorCargaCrudos dom = new DominioTxErrorCargaCrudos();  
 dom.setCount(businessTxErrorCargaCrudos.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorCargaCrudos.count: "+ex);  
 DominioTxErrorCargaCrudos dom = new DominioTxErrorCargaCrudos();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxErrorCargaCrudos existsById(@RequestBody DominioTxErrorCargaCrudos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxErrorCargaCrudos.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxErrorCargaCrudos)businessTxErrorCargaCrudos.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorCargaCrudos.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxErrorFormato.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxErrorFormato;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxErrorFormato;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxErrorFormato")  
public class RESTTxErrorFormato {  
 private static final Logger LOG = LogManager.getLogger(RESTTxErrorFormato.class);  
 @Autowired  
 private IfaceTxErrorFormato businessTxErrorFormato;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxErrorFormato> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxErrorFormato.findAll. . .");  
 List<DominioTxErrorFormato> lista = businessTxErrorFormato.findAll();  
 LOG.info("invoke RESTTxErrorFormato.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorFormato.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxErrorFormato dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxErrorFormato.findAllPagination. . .");  
 response = businessTxErrorFormato.allPagination(dom);  
 LOG.info("invoke RESTTxErrorFormato.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorFormato.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxErrorFormato.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxErrorFormato save(@RequestBody DominioTxErrorFormato dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxErrorFormato.save. . .");  
 LOG.info("dom.getErrorFormatoId(): "+dom.getErrorFormatoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxErrorFormato)businessTxErrorFormato.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorFormato.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxErrorFormato update(@RequestBody DominioTxErrorFormato dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxErrorFormato.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxErrorFormato)businessTxErrorFormato.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorFormato.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxErrorFormato delete(@RequestBody DominioTxErrorFormato dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxErrorFormato.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxErrorFormato)businessTxErrorFormato.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorFormato.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxErrorFormato findById(@RequestBody DominioTxErrorFormato dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxErrorFormato.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxErrorFormato)businessTxErrorFormato.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorFormato.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxErrorFormato count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxErrorFormato.count. . .");  
 DominioTxErrorFormato dom = new DominioTxErrorFormato();  
 dom.setCount(businessTxErrorFormato.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorFormato.count: "+ex);  
 DominioTxErrorFormato dom = new DominioTxErrorFormato();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxErrorFormato existsById(@RequestBody DominioTxErrorFormato dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxErrorFormato.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxErrorFormato)businessTxErrorFormato.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxErrorFormato.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxEscenario.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxEscenario;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxEscenario;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxEscenario")  
public class RESTTxEscenario {  
 private static final Logger LOG = LogManager.getLogger(RESTTxEscenario.class);  
 @Autowired  
 private IfaceTxEscenario businessTxEscenario;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxEscenario> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenario.findAll. . .");  
 List<DominioTxEscenario> lista = businessTxEscenario.findAll();  
 LOG.info("invoke RESTTxEscenario.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenario.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxEscenario dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxEscenario.findAllPagination. . .");  
 response = businessTxEscenario.allPagination(dom);  
 LOG.info("invoke RESTTxEscenario.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenario.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxEscenario.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenario save(@RequestBody DominioTxEscenario dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenario.save. . .");  
 LOG.info("dom.getEscenarioId(): "+dom.getEscenarioId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEscenario)businessTxEscenario.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenario.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenario update(@RequestBody DominioTxEscenario dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenario.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEscenario)businessTxEscenario.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenario.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenario delete(@RequestBody DominioTxEscenario dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenario.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEscenario)businessTxEscenario.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenario.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenario findById(@RequestBody DominioTxEscenario dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenario.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEscenario)businessTxEscenario.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenario.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenario count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenario.count. . .");  
 DominioTxEscenario dom = new DominioTxEscenario();  
 dom.setCount(businessTxEscenario.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenario.count: "+ex);  
 DominioTxEscenario dom = new DominioTxEscenario();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenario existsById(@RequestBody DominioTxEscenario dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenario.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxEscenario)businessTxEscenario.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenario.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxEscenariosBitacora.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxEscenariosBitacora;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxEscenariosBitacora;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxEscenariosBitacora")  
public class RESTTxEscenariosBitacora {  
 private static final Logger LOG = LogManager.getLogger(RESTTxEscenariosBitacora.class);  
 @Autowired  
 private IfaceTxEscenariosBitacora businessTxEscenariosBitacora;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxEscenariosBitacora> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenariosBitacora.findAll. . .");  
 List<DominioTxEscenariosBitacora> lista = businessTxEscenariosBitacora.findAll();  
 LOG.info("invoke RESTTxEscenariosBitacora.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenariosBitacora.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxEscenariosBitacora dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxEscenariosBitacora.findAllPagination. . .");  
 response = businessTxEscenariosBitacora.allPagination(dom);  
 LOG.info("invoke RESTTxEscenariosBitacora.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenariosBitacora.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxEscenariosBitacora.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenariosBitacora save(@RequestBody DominioTxEscenariosBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenariosBitacora.save. . .");  
 LOG.info("dom.getEscenarioBitacoraId(): "+dom.getEscenarioBitacoraId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEscenariosBitacora)businessTxEscenariosBitacora.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenariosBitacora.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenariosBitacora update(@RequestBody DominioTxEscenariosBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenariosBitacora.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEscenariosBitacora)businessTxEscenariosBitacora.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenariosBitacora.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenariosBitacora delete(@RequestBody DominioTxEscenariosBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenariosBitacora.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEscenariosBitacora)businessTxEscenariosBitacora.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenariosBitacora.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenariosBitacora findById(@RequestBody DominioTxEscenariosBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenariosBitacora.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEscenariosBitacora)businessTxEscenariosBitacora.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenariosBitacora.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenariosBitacora count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenariosBitacora.count. . .");  
 DominioTxEscenariosBitacora dom = new DominioTxEscenariosBitacora();  
 dom.setCount(businessTxEscenariosBitacora.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenariosBitacora.count: "+ex);  
 DominioTxEscenariosBitacora dom = new DominioTxEscenariosBitacora();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenariosBitacora existsById(@RequestBody DominioTxEscenariosBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenariosBitacora.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxEscenariosBitacora)businessTxEscenariosBitacora.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenariosBitacora.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxEscenarioVariable.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxEscenarioVariable;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxEscenarioVariable;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxEscenarioVariable")  
public class RESTTxEscenarioVariable {  
 private static final Logger LOG = LogManager.getLogger(RESTTxEscenarioVariable.class);  
 @Autowired  
 private IfaceTxEscenarioVariable businessTxEscenarioVariable;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxEscenarioVariable> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenarioVariable.findAll. . .");  
 List<DominioTxEscenarioVariable> lista = businessTxEscenarioVariable.findAll();  
 LOG.info("invoke RESTTxEscenarioVariable.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenarioVariable.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxEscenarioVariable dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxEscenarioVariable.findAllPagination. . .");  
 response = businessTxEscenarioVariable.allPagination(dom);  
 LOG.info("invoke RESTTxEscenarioVariable.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenarioVariable.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxEscenarioVariable.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenarioVariable save(@RequestBody DominioTxEscenarioVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenarioVariable.save. . .");  
 LOG.info("dom.getEscenarioVariableId(): "+dom.getEscenarioVariableId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEscenarioVariable)businessTxEscenarioVariable.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenarioVariable.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenarioVariable update(@RequestBody DominioTxEscenarioVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenarioVariable.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEscenarioVariable)businessTxEscenarioVariable.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenarioVariable.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenarioVariable delete(@RequestBody DominioTxEscenarioVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenarioVariable.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEscenarioVariable)businessTxEscenarioVariable.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenarioVariable.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenarioVariable findById(@RequestBody DominioTxEscenarioVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenarioVariable.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEscenarioVariable)businessTxEscenarioVariable.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenarioVariable.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenarioVariable count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenarioVariable.count. . .");  
 DominioTxEscenarioVariable dom = new DominioTxEscenarioVariable();  
 dom.setCount(businessTxEscenarioVariable.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenarioVariable.count: "+ex);  
 DominioTxEscenarioVariable dom = new DominioTxEscenarioVariable();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxEscenarioVariable existsById(@RequestBody DominioTxEscenarioVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEscenarioVariable.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxEscenarioVariable)businessTxEscenarioVariable.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxEscenarioVariable.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxEvento.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceTxEvento;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxBitacora;  
import mx.mbi.dgc.dominio.DominioTxEvento;  
import mx.mbi.dgc.dominio.negocio.bitacora.DominioBusquedaEventos;  
import mx.mbi.dgc.dominio.negocio.bitacora.DominioEventoBitacora;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxEvento")  
public class RESTTxEvento {  
 private static final Logger LOG = LogManager.getLogger(RESTTxEvento.class);  
 @Autowired  
 private IfaceTxEvento businessTxEvento;  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAll", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioTxEvento> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxEvento.findAll. . .");  
 List<DominioTxEvento> lista = businessTxEvento.findAll();  
 LOG.info("invoke RESTTxEvento.findAll size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEvento.findAll: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findByBitacora", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioTxEvento> findByBitacora(@RequestBody DominioBusquedaEventos dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEvento.findByBitacora. . .");  
 List<DominioTxEvento> lista = businessTxEvento.findByBitacora(dom);  
 LOG.info("invoke RESTTxEvento.findByBitacora size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEvento.findByBitacora: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAllPagination", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioRespuesta findAllPagination(@RequestBody DominioTxEvento dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxEvento.findAllPagination. . .");  
 response = businessTxEvento.allPagination(dom);  
 LOG.info("invoke RESTTxEvento.findAllPagination size lista return: " + response.getData().size());  
 return response;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEvento.findAllPagination: " + ex);  
 response.setMensaje("Error in invoke RESTTxEvento.findAllPagination: " + ex);  
 return response;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/save", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxEvento save(@RequestBody DominioTxEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEvento.save. . .");  
 LOG.info("dom.getEventoId(): " + dom.getEventoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEvento) businessTxEvento.save(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEvento.save: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/guardarEvento", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxEvento guardarEvento(@RequestBody DominioEventoBitacora dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEvento.guardarEvento. . .");  
 //LOG.info("dom.getEventoId(): "+dom.getEventoId());  
 DominioTxEvento res = businessTxEvento.guardarEvento(dom);  
 res.setStatus(false);  
 res.setCode(500);  
 return res;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEvento.guardarEvento: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return null;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/update", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxEvento update(@RequestBody DominioTxEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEvento.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEvento) businessTxEvento.update(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEvento.update: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/delete", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxEvento delete(@RequestBody DominioTxEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEvento.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEvento) businessTxEvento.delete(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEvento.delete: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxEvento findById(@RequestBody DominioTxEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEvento.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEvento) businessTxEvento.findById(dom);  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 LOG.error("Error in invoke RESTTxEvento.findById: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/count", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxEvento count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxEvento.count. . .");  
 DominioTxEvento dom = new DominioTxEvento();  
 dom.setCount(businessTxEvento.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEvento.count: " + ex);  
 DominioTxEvento dom = new DominioTxEvento();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/existsById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxEvento existsById(@RequestBody DominioTxEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEvento.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxEvento) businessTxEvento.existsById(dom);  
 if (dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEvento.existsById: " + ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxEventoDetalles.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import mx.mbi.dgc.business.IfaceTxEventoDetalles;  
import mx.mbi.dgc.dominio.DominioTxEvento;  
import mx.mbi.dgc.dominio.DominioTxEventoDetalles;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxEventoDetalles")  
public class RESTTxEventoDetalles {  
 private static final Logger LOG = LogManager.getLogger(RESTTxEventoDetalles.class);  
 @Autowired  
 private IfaceTxEventoDetalles businessTxEventoDetalles;  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findAll", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioTxEventoDetalles> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxEventoDetalles.findAll. . .");  
 List<DominioTxEventoDetalles> lista = businessTxEventoDetalles.findAll();  
 LOG.info("invoke RESTTxEventoDetalles.findAll size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEventoDetalles.findAll: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findByEvento", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 List<DominioTxEventoDetalles> findByEvento(@RequestBody DominioTxEvento dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEventoDetalles.findByEvento. . .");  
 List<DominioTxEventoDetalles> lista = businessTxEventoDetalles.findByTxEvento(dom);  
 LOG.info("invoke RESTTxEventoDetalles.findByBitacora size lista return: " + lista.size());  
 return lista;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEventoDetalles.findByBitfindByEventoacora: " + ex);  
 return new ArrayList<>();  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/save", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxEventoDetalles save(@RequestBody DominioTxEventoDetalles dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEventoDetalles.save. . .");  
 LOG.info("dom.getEventoDetallesId(): " + dom.getEventoDetalleId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEventoDetalles) businessTxEventoDetalles.save(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEventoDetalles.save: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
  
 @RequestMapping(method = RequestMethod.POST, path = "/update", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxEventoDetalles update(@RequestBody DominioTxEventoDetalles dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEventoDetalles.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEventoDetalles) businessTxEventoDetalles.update(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEventoDetalles.update: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/delete", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxEventoDetalles delete(@RequestBody DominioTxEventoDetalles dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEventoDetalles.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEventoDetalles) businessTxEventoDetalles.delete(dom);  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEventoDetalles.delete: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/findById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxEventoDetalles findById(@RequestBody DominioTxEventoDetalles dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEventoDetalles.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxEventoDetalles) businessTxEventoDetalles.findById(dom);  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 LOG.error("Error in invoke RESTTxEventoDetalles.findById: " + ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/count", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxEventoDetalles count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxEventoDetalles.count. . .");  
 DominioTxEventoDetalles dom = new DominioTxEventoDetalles();  
 dom.setCount(businessTxEventoDetalles.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEventoDetalles.count: " + ex);  
 DominioTxEventoDetalles dom = new DominioTxEventoDetalles();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
  
 @RequestMapping(method = RequestMethod.POST, path = "/existsById", consumes = "application/json",  
 produces = "application/json")  
 public @ResponseBody  
 DominioTxEventoDetalles existsById(@RequestBody DominioTxEventoDetalles dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxEventoDetalles.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxEventoDetalles) businessTxEventoDetalles.existsById(dom);  
 if (dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch (Exception ex) {  
 LOG.error("Error in invoke RESTTxEventoDetalles.existsById: " + ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxFacturaUuids.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxFacturaUuids;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxFacturaUuids;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxFacturaUuids")  
public class RESTTxFacturaUuids {  
 private static final Logger LOG = LogManager.getLogger(RESTTxFacturaUuids.class);  
 @Autowired  
 private IfaceTxFacturaUuids businessTxFacturaUuids;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxFacturaUuids> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxFacturaUuids.findAll. . .");  
 List<DominioTxFacturaUuids> lista = businessTxFacturaUuids.findAll();  
 LOG.info("invoke RESTTxFacturaUuids.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFacturaUuids.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxFacturaUuids dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxFacturaUuids.findAllPagination. . .");  
 response = businessTxFacturaUuids.allPagination(dom);  
 LOG.info("invoke RESTTxFacturaUuids.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFacturaUuids.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxFacturaUuids.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxFacturaUuids save(@RequestBody DominioTxFacturaUuids dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxFacturaUuids.save. . .");  
 LOG.info("dom.getFacturaUuidId(): "+dom.getFacturaUuidId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxFacturaUuids)businessTxFacturaUuids.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFacturaUuids.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxFacturaUuids update(@RequestBody DominioTxFacturaUuids dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxFacturaUuids.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxFacturaUuids)businessTxFacturaUuids.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFacturaUuids.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxFacturaUuids delete(@RequestBody DominioTxFacturaUuids dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxFacturaUuids.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxFacturaUuids)businessTxFacturaUuids.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFacturaUuids.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxFacturaUuids findById(@RequestBody DominioTxFacturaUuids dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxFacturaUuids.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxFacturaUuids)businessTxFacturaUuids.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFacturaUuids.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxFacturaUuids count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxFacturaUuids.count. . .");  
 DominioTxFacturaUuids dom = new DominioTxFacturaUuids();  
 dom.setCount(businessTxFacturaUuids.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFacturaUuids.count: "+ex);  
 DominioTxFacturaUuids dom = new DominioTxFacturaUuids();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxFacturaUuids existsById(@RequestBody DominioTxFacturaUuids dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxFacturaUuids.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxFacturaUuids)businessTxFacturaUuids.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFacturaUuids.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxFormulas.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxFormulas;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxFormulas;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxFormulas")  
public class RESTTxFormulas {  
 private static final Logger LOG = LogManager.getLogger(RESTTxFormulas.class);  
 @Autowired  
 private IfaceTxFormulas businessTxFormulas;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxFormulas> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxFormulas.findAll. . .");  
 List<DominioTxFormulas> lista = businessTxFormulas.findAll();  
 LOG.info("invoke RESTTxFormulas.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFormulas.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxFormulas dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxFormulas.findAllPagination. . .");  
 response = businessTxFormulas.allPagination(dom);  
 LOG.info("invoke RESTTxFormulas.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFormulas.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxFormulas.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxFormulas save(@RequestBody DominioTxFormulas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxFormulas.save. . .");  
 LOG.info("dom.getFormulaId(): "+dom.getFormulaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxFormulas)businessTxFormulas.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFormulas.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxFormulas update(@RequestBody DominioTxFormulas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxFormulas.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxFormulas)businessTxFormulas.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFormulas.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxFormulas delete(@RequestBody DominioTxFormulas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxFormulas.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxFormulas)businessTxFormulas.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFormulas.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxFormulas findById(@RequestBody DominioTxFormulas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxFormulas.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxFormulas)businessTxFormulas.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFormulas.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxFormulas count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxFormulas.count. . .");  
 DominioTxFormulas dom = new DominioTxFormulas();  
 dom.setCount(businessTxFormulas.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFormulas.count: "+ex);  
 DominioTxFormulas dom = new DominioTxFormulas();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxFormulas existsById(@RequestBody DominioTxFormulas dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxFormulas.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxFormulas)businessTxFormulas.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxFormulas.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxLogErrorCarga.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxLogErrorCarga;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxLogErrorCarga;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxLogErrorCarga")  
public class RESTTxLogErrorCarga {  
 private static final Logger LOG = LogManager.getLogger(RESTTxLogErrorCarga.class);  
 @Autowired  
 private IfaceTxLogErrorCarga businessTxLogErrorCarga;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxLogErrorCarga> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxLogErrorCarga.findAll. . .");  
 List<DominioTxLogErrorCarga> lista = businessTxLogErrorCarga.findAll();  
 LOG.info("invoke RESTTxLogErrorCarga.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxLogErrorCarga.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxLogErrorCarga dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxLogErrorCarga.findAllPagination. . .");  
 response = businessTxLogErrorCarga.allPagination(dom);  
 LOG.info("invoke RESTTxLogErrorCarga.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxLogErrorCarga.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxLogErrorCarga.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxLogErrorCarga save(@RequestBody DominioTxLogErrorCarga dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxLogErrorCarga.save. . .");  
 LOG.info("dom.getLogErrorCargaId(): "+dom.getLogErrorCargaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxLogErrorCarga)businessTxLogErrorCarga.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxLogErrorCarga.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxLogErrorCarga update(@RequestBody DominioTxLogErrorCarga dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxLogErrorCarga.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxLogErrorCarga)businessTxLogErrorCarga.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxLogErrorCarga.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxLogErrorCarga delete(@RequestBody DominioTxLogErrorCarga dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxLogErrorCarga.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxLogErrorCarga)businessTxLogErrorCarga.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxLogErrorCarga.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxLogErrorCarga findById(@RequestBody DominioTxLogErrorCarga dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxLogErrorCarga.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxLogErrorCarga)businessTxLogErrorCarga.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxLogErrorCarga.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxLogErrorCarga count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxLogErrorCarga.count. . .");  
 DominioTxLogErrorCarga dom = new DominioTxLogErrorCarga();  
 dom.setCount(businessTxLogErrorCarga.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxLogErrorCarga.count: "+ex);  
 DominioTxLogErrorCarga dom = new DominioTxLogErrorCarga();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxLogErrorCarga existsById(@RequestBody DominioTxLogErrorCarga dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxLogErrorCarga.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxLogErrorCarga)businessTxLogErrorCarga.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxLogErrorCarga.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxPerfilesVariable.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxPerfilesVariable;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxPerfilesVariable;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxPerfilesVariable")  
public class RESTTxPerfilesVariable {  
 private static final Logger LOG = LogManager.getLogger(RESTTxPerfilesVariable.class);  
 @Autowired  
 private IfaceTxPerfilesVariable businessTxPerfilesVariable;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxPerfilesVariable> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxPerfilesVariable.findAll. . .");  
 List<DominioTxPerfilesVariable> lista = businessTxPerfilesVariable.findAll();  
 LOG.info("invoke RESTTxPerfilesVariable.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxPerfilesVariable.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxPerfilesVariable dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxPerfilesVariable.findAllPagination. . .");  
 response = businessTxPerfilesVariable.allPagination(dom);  
 LOG.info("invoke RESTTxPerfilesVariable.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxPerfilesVariable.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxPerfilesVariable.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxPerfilesVariable save(@RequestBody DominioTxPerfilesVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxPerfilesVariable.save. . .");  
 LOG.info("dom.getPerfilId(): "+dom.getPerfilId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxPerfilesVariable)businessTxPerfilesVariable.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxPerfilesVariable.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxPerfilesVariable update(@RequestBody DominioTxPerfilesVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxPerfilesVariable.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxPerfilesVariable)businessTxPerfilesVariable.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxPerfilesVariable.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxPerfilesVariable delete(@RequestBody DominioTxPerfilesVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxPerfilesVariable.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxPerfilesVariable)businessTxPerfilesVariable.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxPerfilesVariable.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxPerfilesVariable findById(@RequestBody DominioTxPerfilesVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxPerfilesVariable.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxPerfilesVariable)businessTxPerfilesVariable.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxPerfilesVariable.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxPerfilesVariable count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxPerfilesVariable.count. . .");  
 DominioTxPerfilesVariable dom = new DominioTxPerfilesVariable();  
 dom.setCount(businessTxPerfilesVariable.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxPerfilesVariable.count: "+ex);  
 DominioTxPerfilesVariable dom = new DominioTxPerfilesVariable();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxPerfilesVariable existsById(@RequestBody DominioTxPerfilesVariable dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxPerfilesVariable.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxPerfilesVariable)businessTxPerfilesVariable.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxPerfilesVariable.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxProduccion.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxProduccion;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxProduccion;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxProduccion")  
public class RESTTxProduccion {  
 private static final Logger LOG = LogManager.getLogger(RESTTxProduccion.class);  
 @Autowired  
 private IfaceTxProduccion businessTxProduccion;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxProduccion> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxProduccion.findAll. . .");  
 List<DominioTxProduccion> lista = businessTxProduccion.findAll();  
 LOG.info("invoke RESTTxProduccion.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxProduccion.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxProduccion dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxProduccion.findAllPagination. . .");  
 response = businessTxProduccion.allPagination(dom);  
 LOG.info("invoke RESTTxProduccion.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxProduccion.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxProduccion.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxProduccion save(@RequestBody DominioTxProduccion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxProduccion.save. . .");  
 LOG.info("dom.getProduccionId(): "+dom.getProduccionId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxProduccion)businessTxProduccion.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxProduccion.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxProduccion update(@RequestBody DominioTxProduccion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxProduccion.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxProduccion)businessTxProduccion.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxProduccion.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxProduccion delete(@RequestBody DominioTxProduccion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxProduccion.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxProduccion)businessTxProduccion.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxProduccion.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxProduccion findById(@RequestBody DominioTxProduccion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxProduccion.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxProduccion)businessTxProduccion.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxProduccion.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxProduccion count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxProduccion.count. . .");  
 DominioTxProduccion dom = new DominioTxProduccion();  
 dom.setCount(businessTxProduccion.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxProduccion.count: "+ex);  
 DominioTxProduccion dom = new DominioTxProduccion();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxProduccion existsById(@RequestBody DominioTxProduccion dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxProduccion.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxProduccion)businessTxProduccion.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxProduccion.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxRolPermiso.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxRolPermiso;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxRolPermiso;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxRolPermiso")  
public class RESTTxRolPermiso {  
 private static final Logger LOG = LogManager.getLogger(RESTTxRolPermiso.class);  
 @Autowired  
 private IfaceTxRolPermiso businessTxRolPermiso;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxRolPermiso> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxRolPermiso.findAll. . .");  
 List<DominioTxRolPermiso> lista = businessTxRolPermiso.findAll();  
 LOG.info("invoke RESTTxRolPermiso.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxRolPermiso.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxRolPermiso dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxRolPermiso.findAllPagination. . .");  
 response = businessTxRolPermiso.allPagination(dom);  
 LOG.info("invoke RESTTxRolPermiso.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxRolPermiso.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxRolPermiso.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxRolPermiso save(@RequestBody DominioTxRolPermiso dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxRolPermiso.save. . .");  
 LOG.info("dom.getRolPermisoId(): "+dom.getRolPermisoId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxRolPermiso)businessTxRolPermiso.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxRolPermiso.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxRolPermiso update(@RequestBody DominioTxRolPermiso dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxRolPermiso.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxRolPermiso)businessTxRolPermiso.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxRolPermiso.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxRolPermiso delete(@RequestBody DominioTxRolPermiso dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxRolPermiso.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxRolPermiso)businessTxRolPermiso.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxRolPermiso.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxRolPermiso findById(@RequestBody DominioTxRolPermiso dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxRolPermiso.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxRolPermiso)businessTxRolPermiso.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxRolPermiso.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxRolPermiso count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxRolPermiso.count. . .");  
 DominioTxRolPermiso dom = new DominioTxRolPermiso();  
 dom.setCount(businessTxRolPermiso.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxRolPermiso.count: "+ex);  
 DominioTxRolPermiso dom = new DominioTxRolPermiso();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxRolPermiso existsById(@RequestBody DominioTxRolPermiso dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxRolPermiso.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxRolPermiso)businessTxRolPermiso.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxRolPermiso.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxTrazabilidad.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxTrazabilidad;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxTrazabilidad;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxTrazabilidad")  
public class RESTTxTrazabilidad {  
 private static final Logger LOG = LogManager.getLogger(RESTTxTrazabilidad.class);  
 @Autowired  
 private IfaceTxTrazabilidad businessTxTrazabilidad;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxTrazabilidad> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxTrazabilidad.findAll. . .");  
 List<DominioTxTrazabilidad> lista = businessTxTrazabilidad.findAll();  
 LOG.info("invoke RESTTxTrazabilidad.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazabilidad.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxTrazabilidad dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxTrazabilidad.findAllPagination. . .");  
 response = businessTxTrazabilidad.allPagination(dom);  
 LOG.info("invoke RESTTxTrazabilidad.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazabilidad.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxTrazabilidad.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxTrazabilidad save(@RequestBody DominioTxTrazabilidad dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxTrazabilidad.save. . .");  
 LOG.info("dom.getTrazabilidadId(): "+dom.getTrazabilidadId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxTrazabilidad)businessTxTrazabilidad.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazabilidad.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxTrazabilidad update(@RequestBody DominioTxTrazabilidad dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxTrazabilidad.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxTrazabilidad)businessTxTrazabilidad.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazabilidad.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxTrazabilidad delete(@RequestBody DominioTxTrazabilidad dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxTrazabilidad.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxTrazabilidad)businessTxTrazabilidad.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazabilidad.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxTrazabilidad findById(@RequestBody DominioTxTrazabilidad dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxTrazabilidad.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxTrazabilidad)businessTxTrazabilidad.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazabilidad.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxTrazabilidad count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxTrazabilidad.count. . .");  
 DominioTxTrazabilidad dom = new DominioTxTrazabilidad();  
 dom.setCount(businessTxTrazabilidad.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazabilidad.count: "+ex);  
 DominioTxTrazabilidad dom = new DominioTxTrazabilidad();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxTrazabilidad existsById(@RequestBody DominioTxTrazabilidad dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxTrazabilidad.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxTrazabilidad)businessTxTrazabilidad.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazabilidad.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxTrazaQuery.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxTrazaQuery;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxTrazaQuery;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxTrazaQuery")  
public class RESTTxTrazaQuery {  
 private static final Logger LOG = LogManager.getLogger(RESTTxTrazaQuery.class);  
 @Autowired  
 private IfaceTxTrazaQuery businessTxTrazaQuery;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxTrazaQuery> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxTrazaQuery.findAll. . .");  
 List<DominioTxTrazaQuery> lista = businessTxTrazaQuery.findAll();  
 LOG.info("invoke RESTTxTrazaQuery.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazaQuery.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxTrazaQuery dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxTrazaQuery.findAllPagination. . .");  
 response = businessTxTrazaQuery.allPagination(dom);  
 LOG.info("invoke RESTTxTrazaQuery.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazaQuery.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxTrazaQuery.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxTrazaQuery save(@RequestBody DominioTxTrazaQuery dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxTrazaQuery.save. . .");  
 LOG.info("dom.getTrazaQueryId(): "+dom.getTrazaQueryId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxTrazaQuery)businessTxTrazaQuery.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazaQuery.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxTrazaQuery update(@RequestBody DominioTxTrazaQuery dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxTrazaQuery.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxTrazaQuery)businessTxTrazaQuery.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazaQuery.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxTrazaQuery delete(@RequestBody DominioTxTrazaQuery dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxTrazaQuery.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxTrazaQuery)businessTxTrazaQuery.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazaQuery.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxTrazaQuery findById(@RequestBody DominioTxTrazaQuery dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxTrazaQuery.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxTrazaQuery)businessTxTrazaQuery.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazaQuery.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxTrazaQuery count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxTrazaQuery.count. . .");  
 DominioTxTrazaQuery dom = new DominioTxTrazaQuery();  
 dom.setCount(businessTxTrazaQuery.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazaQuery.count: "+ex);  
 DominioTxTrazaQuery dom = new DominioTxTrazaQuery();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxTrazaQuery existsById(@RequestBody DominioTxTrazaQuery dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxTrazaQuery.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxTrazaQuery)businessTxTrazaQuery.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxTrazaQuery.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxUsuario.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxUsuario;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxUsuario;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxUsuario")  
public class RESTTxUsuario {  
 private static final Logger LOG = LogManager.getLogger(RESTTxUsuario.class);  
 @Autowired  
 private IfaceTxUsuario businessTxUsuario;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxUsuario> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuario.findAll. . .");  
 List<DominioTxUsuario> lista = businessTxUsuario.findAll();  
 LOG.info("invoke RESTTxUsuario.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuario.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxUsuario dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxUsuario.findAllPagination. . .");  
 response = businessTxUsuario.allPagination(dom);  
 LOG.info("invoke RESTTxUsuario.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuario.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxUsuario.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuario save(@RequestBody DominioTxUsuario dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuario.save. . .");  
 LOG.info("dom.getUsuarioId(): "+dom.getUsuarioId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxUsuario)businessTxUsuario.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuario.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuario update(@RequestBody DominioTxUsuario dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuario.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxUsuario)businessTxUsuario.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuario.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuario delete(@RequestBody DominioTxUsuario dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuario.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxUsuario)businessTxUsuario.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuario.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuario findById(@RequestBody DominioTxUsuario dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuario.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxUsuario)businessTxUsuario.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuario.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuario count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuario.count. . .");  
 DominioTxUsuario dom = new DominioTxUsuario();  
 dom.setCount(businessTxUsuario.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuario.count: "+ex);  
 DominioTxUsuario dom = new DominioTxUsuario();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuario existsById(@RequestBody DominioTxUsuario dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuario.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxUsuario)businessTxUsuario.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuario.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxUsuarioPlanta.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxUsuarioPlanta;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxUsuarioPlanta;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxUsuarioPlanta")  
public class RESTTxUsuarioPlanta {  
 private static final Logger LOG = LogManager.getLogger(RESTTxUsuarioPlanta.class);  
 @Autowired  
 private IfaceTxUsuarioPlanta businessTxUsuarioPlanta;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxUsuarioPlanta> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuarioPlanta.findAll. . .");  
 List<DominioTxUsuarioPlanta> lista = businessTxUsuarioPlanta.findAll();  
 LOG.info("invoke RESTTxUsuarioPlanta.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioPlanta.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxUsuarioPlanta dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxUsuarioPlanta.findAllPagination. . .");  
 response = businessTxUsuarioPlanta.allPagination(dom);  
 LOG.info("invoke RESTTxUsuarioPlanta.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioPlanta.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxUsuarioPlanta.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuarioPlanta save(@RequestBody DominioTxUsuarioPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuarioPlanta.save. . .");  
 LOG.info("dom.getUsuarioPlantaId(): "+dom.getUsuarioPlantaId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxUsuarioPlanta)businessTxUsuarioPlanta.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioPlanta.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuarioPlanta update(@RequestBody DominioTxUsuarioPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuarioPlanta.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxUsuarioPlanta)businessTxUsuarioPlanta.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioPlanta.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuarioPlanta delete(@RequestBody DominioTxUsuarioPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuarioPlanta.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxUsuarioPlanta)businessTxUsuarioPlanta.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioPlanta.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuarioPlanta findById(@RequestBody DominioTxUsuarioPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuarioPlanta.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxUsuarioPlanta)businessTxUsuarioPlanta.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioPlanta.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuarioPlanta count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuarioPlanta.count. . .");  
 DominioTxUsuarioPlanta dom = new DominioTxUsuarioPlanta();  
 dom.setCount(businessTxUsuarioPlanta.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioPlanta.count: "+ex);  
 DominioTxUsuarioPlanta dom = new DominioTxUsuarioPlanta();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuarioPlanta existsById(@RequestBody DominioTxUsuarioPlanta dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuarioPlanta.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxUsuarioPlanta)businessTxUsuarioPlanta.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioPlanta.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\controller\RESTTxUsuarioRol.java

package mx.mbi.dgc.microservices.controller;  
  
import java.util.ArrayList;  
  
import mx.mbi.dgc.business.IfaceTxUsuarioRol;  
import mx.mbi.dgc.dominio.DominioRespuesta;  
import mx.mbi.dgc.dominio.DominioTxUsuarioRol;  
import java.util.List;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.RestController;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
@RestController  
@RequestMapping(path = "/servicesREST/TxUsuarioRol")  
public class RESTTxUsuarioRol {  
 private static final Logger LOG = LogManager.getLogger(RESTTxUsuarioRol.class);  
 @Autowired  
 private IfaceTxUsuarioRol businessTxUsuarioRol;  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAll",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody List<DominioTxUsuarioRol> findAll() throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuarioRol.findAll. . .");  
 List<DominioTxUsuarioRol> lista = businessTxUsuarioRol.findAll();  
 LOG.info("invoke RESTTxUsuarioRol.findAll size lista return: "+lista.size());  
 return lista;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioRol.findAll: "+ex);  
 return new ArrayList<>();  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findAllPagination",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioRespuesta findAllPagination(@RequestBody DominioTxUsuarioRol dom) throws Exception {  
 DominioRespuesta response = new DominioRespuesta();  
 response.setCodigo(500);  
 response.setMensaje("error");  
 try {  
 LOG.info("invoke RESTTxUsuarioRol.findAllPagination. . .");  
 response = businessTxUsuarioRol.allPagination(dom);  
 LOG.info("invoke RESTTxUsuarioRol.findAllPagination size lista return: "+response.getData().size());  
 return response;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioRol.findAllPagination: "+ex);  
 response.setMensaje("Error in invoke RESTTxUsuarioRol.findAllPagination: "+ex);  
 return response;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/save",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuarioRol save(@RequestBody DominioTxUsuarioRol dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuarioRol.save. . .");  
 LOG.info("dom.getUsuarioRolId(): "+dom.getUsuarioRolId());  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxUsuarioRol)businessTxUsuarioRol.save(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioRol.save: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/update",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuarioRol update(@RequestBody DominioTxUsuarioRol dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuarioRol.update. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxUsuarioRol)businessTxUsuarioRol.update(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioRol.update: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/delete",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuarioRol delete(@RequestBody DominioTxUsuarioRol dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuarioRol.delete. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxUsuarioRol)businessTxUsuarioRol.delete(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioRol.delete: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/findById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuarioRol findById(@RequestBody DominioTxUsuarioRol dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuarioRol.findById. . .");  
 dom.setStatus(false);  
 dom.setCode(500);  
 return (DominioTxUsuarioRol)businessTxUsuarioRol.findById(dom);  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioRol.findById: "+ex);  
 dom.setStatus(false);  
 dom.setCode(500);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/count",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuarioRol count() throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuarioRol.count. . .");  
 DominioTxUsuarioRol dom = new DominioTxUsuarioRol();  
 dom.setCount(businessTxUsuarioRol.count());  
 dom.setStatus(true);  
 dom.setCode(200);  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioRol.count: "+ex);  
 DominioTxUsuarioRol dom = new DominioTxUsuarioRol();  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
   
 @RequestMapping(  
 method = RequestMethod.POST, path = "/existsById",  
 consumes = "application/json", produces = "application/json"  
 )  
 public @ResponseBody DominioTxUsuarioRol existsById(@RequestBody DominioTxUsuarioRol dom) throws Exception {  
 try {  
 LOG.info("invoke RESTTxUsuarioRol.existsById. . .");  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom = (DominioTxUsuarioRol)businessTxUsuarioRol.existsById(dom);  
 if(dom.isFound()) {  
 dom.setStatus(true);  
 dom.setCode(200);  
 }  
 return dom;  
 } catch(Exception ex) {  
 LOG.error("Error in invoke RESTTxUsuarioRol.existsById: "+ex);  
 dom.setFound(false);  
 dom.setStatus(false);  
 dom.setCode(500);  
 dom.setCount(-1);  
 return dom;  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\microservices\MicroServicesWebAppJR.java

package mx.mbi.dgc.microservices;  
  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.boot.autoconfigure.domain.EntityScan;  
import org.springframework.context.annotation.ComponentScan;  
  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import org.springframework.boot.Banner;  
import org.springframework.boot.builder.SpringApplicationBuilder;  
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;  
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;  
import org.springframework.web.WebApplicationInitializer;  
  
@SpringBootApplication  
@ComponentScan(basePackages = "mx.mbi.dgc")  
@EnableJpaRepositories(basePackages="mx.mbi.dgc.dao")  
@EntityScan(basePackages = "mx.mbi.dgc.jpa.entity")  
public class MicroServicesWebAppJR extends SpringBootServletInitializer implements WebApplicationInitializer {  
 private static final Logger LOG = LogManager.getLogger(MicroServicesWebAppJR.class);  
  
 @Override  
 protected SpringApplicationBuilder configure(SpringApplicationBuilder builder) {  
 LOG.info("invoke MicroServicesWebAppJR.configure. . .");  
 return configureApplication(builder);  
 }  
  
 private static SpringApplicationBuilder configureApplication(SpringApplicationBuilder builder) {  
 LOG.info("invoke MicroServicesWebAppJR.configureApplication. . .");  
 return builder.sources(MicroServicesWebAppJR.class).bannerMode(Banner.Mode.OFF);  
 }  
   
 /\*\*  
 \* Método principal que inicializa el servidor y aplicativo web con Spring Boot.  
 \*  
 \* @param args  
 \*/  
 public static void main(String[] args) {  
 LOG.info("invoke MicroServicesWebAppJR.main. . .");  
 configureApplication(new SpringApplicationBuilder()).run(args);  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\utils\cantidadesaletras\CantidadesALetras.java

package mx.mbi.dgc.utils.cantidadesaletras;  
  
public class CantidadesALetras {  
 private enum Unidades {  
 Cero(0),  
 Uno(1),  
 Dos(2),  
 Tres(3),  
 Cuatro(4),  
 Cinco(5),  
 Seis(6),  
 Siete(7),  
 Ocho(8),  
 Nueve(9);  
  
 public final int valor; //No se puede cambiar  
  
 Unidades(int valor) {  
 this.valor = valor;  
 }  
  
 /\*\*  
 \* Retorna el nombre con letras de una unidad.  
 \* @param digito representa una cantidad de unidades  
 \* @return la representación escrita con letras  
 \*/  
 static String convertir(int digito) {  
 //Si el número es mayor a 9 o inferior a 0, hubo un error  
 if (digito < 0 || digito > 9) {  
 System.out.println("No se pueden convertir unidades: Digito fuera de rango.");  
 return "-";  
 }  
  
 //Busca el valor que corresponda  
 for (Unidades unidad : Unidades.values()) {  
 if (digito == unidad.valor) {  
 return unidad.toString();  
 }  
 }  
  
 //Esta última rama nunca debería de suceder  
 System.out.println("No se pueden convertir unidades: No se encontro correspondencia.");  
 return "~";  
 }  
  
 @Override  
 public String toString() {  
 if (this.valor == 0) {  
 return "";  
 }  
 return super.toString() + " ";  
 }  
  
 }  
 private enum Decenas{  
 Cero(0),  
 Diez(1),  
 Veinte(2),  
 Treinta(3),  
 Cuarenta(4),  
 Cincuenta(5),  
 Sesenta(6),  
 Setenta(7),  
 Ochenta(8),  
 Noventa(9);  
  
 public final int valor;  
  
 Decenas(int valor) {  
 this.valor = valor;  
 }  
  
 /\*\*  
 \* Retorna el nombre con letras de una decena.  
 \* @param digito representa una cantidad de decenas  
 \* @return la representación escrita con letras  
 \*/  
 static String convertir(int digito) {  
 //Si el número es mayor a 9 o inferior a 0, hubo un error  
 if (digito < 0 || digito > 9) {  
 System.out.println("No se pueden convertir decenas: Digito fuera de rango.");  
 return "-";  
 }  
  
 //Busca el valor que corresponda  
 for (Decenas decena : Decenas.values()) {  
 if (digito == decena.valor) {  
 return decena.toString();  
 }  
 }  
  
 //Esta última rama nunca debería de suceder  
 System.out.println("No se pueden convertir decenas: No se encontro correspondencia.");  
 return "~";  
 }  
  
 @Override  
 public String toString() {  
 if (this.valor == 0) {  
 return "";  
 }  
 return super.toString() + " ";  
 }  
 }  
  
 /\*\*  
 \* Convierte un número entero a letras, por ejemplo 319 a "Trescientos Diecinueve" usando un algoritmo recursivo.  
 \*  
 \* @param numero el número a convertir  
 \* @return el número escrito (importe con letra)  
 \*/  
 public static String convertirImporteALetras(int numero) {  
 return convertirImporteALetras(numero, 0);  
 }  
  
 private static String convertirImporteALetras(int numero, int ordenDeMagnitud) {  
 int digitoEvaluado;  
 int digitoSiguiente;  
 int ordenSiguiente;  
 StringBuilder importeEnLetras = new StringBuilder();  
  
 /\*Cuando el orden de magnitud es múltiplo de 3, se agrupan dos dígitos porque hay  
 algunos valores que se traducen de dos en dos por ejemplo 11, 12, 23 etc. se escriben "once", "doce",  
 "veintitrés" etc.  
 Esto sucede por ejemplo en "once mil millones" o "veintidós mil".\*/  
 if (ordenDeMagnitud % 3 == 0) { //0,3,6,9...  
 digitoEvaluado = numero % 100;  
 digitoSiguiente = numero / 100;  
 ordenSiguiente = 2;  
  
 //Para no escribir "uno" en mil y evitar "Uno Mil", excepto cuando sea algo com 201 000 "doscientos un mil"  
 if (digitoEvaluado != 1 || ((ordenDeMagnitud/3)%2 == 0) || digitoSiguiente != 0) {  
 importeEnLetras.append(convertirValorEntreCeroYCien(digitoEvaluado));  
 }  
  
 if (importeEnLetras.length()>0 && digitoEvaluado % 10 == 1 && ordenDeMagnitud != 0){  
 //Le quita la o a Uno cuando son miles, millones, etc.  
 importeEnLetras.deleteCharAt(importeEnLetras.length() - 2);  
 }  
  
 importeEnLetras.append(potenciaDeMilALetras(digitoEvaluado, ordenDeMagnitud));  
  
 //Para que no diga "un millones" o "un billones" sino "un millon" o "un billon"  
  
 importeEnLetras.delete(importeEnLetras.length() - 3, importeEnLetras.length() - 1);  
 }  
  
 } else { //Cuando el orden de magnitud NO ES múltiplo de 3  
 digitoEvaluado = numero % 10;  
 digitoSiguiente = numero / 10;  
 ordenSiguiente = 1;  
  
 if (ordenDeMagnitud % 3 == 1) {  
 //Decenas  
 importeEnLetras.append(Decenas.convertir(digitoEvaluado));  
 } else {  
 //Cientos  
 importeEnLetras.append(convertirCientos(digitoEvaluado));  
 }  
 }  
  
 //Revisa el dígito siguiente para saber si ya terminó  
 if (digitoSiguiente == 0) {  
 //Terminó, retorna el resultado.  
 return importeEnLetras.toString();  
 } else {  
  
 int ordenDeMagnitudSiguiente = ordenDeMagnitud + ordenSiguiente;  
 StringBuilder resultado = new StringBuilder();  
 resultado.append(convertirImporteALetras(digitoSiguiente,ordenDeMagnitudSiguiente));  
 //Para que cuando sea 100, no diga "ciento", sino sólo "cien"  
 if ((ordenDeMagnitudSiguiente % 3 == 2 || ordenDeMagnitudSiguiente == 0)  
 && digitoEvaluado == 0 && digitoSiguiente % 10 == 1) {  
 resultado.delete(resultado.length() - 3, resultado.length() - 1);  
 }  
  
 resultado.append(importeEnLetras);  
 //Yo <3 recursividad  
 return resultado.toString();  
 }  
 }  
  
 private static String convertirCientos(int digitoEvaluado) {  
 StringBuilder importeEnLetras = new StringBuilder();  
 if (digitoEvaluado == 1) {  
 importeEnLetras.append("Ciento ");  
 } else if (digitoEvaluado == 5) {  
 importeEnLetras.append("Quinientos ");  
 } else if (digitoEvaluado == 7) {  
 importeEnLetras.append("Sete ");  
 } else if (digitoEvaluado == 9) {  
 importeEnLetras.append("Nove ");  
 } else {  
 importeEnLetras.append(Unidades.convertir(digitoEvaluado));  
 }  
 if (importeEnLetras.length() != 0 && digitoEvaluado != 5 && digitoEvaluado != 1) {  
 importeEnLetras.deleteCharAt(importeEnLetras.length() - 1);  
 importeEnLetras.append("cientos ");  
 }  
 return importeEnLetras.toString();  
 }  
  
 private static String potenciaDeMilALetras(int digitoEvaluado, int ordenDeMagnitud) {  
 String ordenDeMagnitudEnLetras = "";  
  
 if (ordenDeMagnitud % 3 != 0) {  
 System.out.println("Orden de magnitud incorrecto: no es multiplo de 3.");  
 return "";  
 }  
  
 int potenciaDeMil = ordenDeMagnitud / 3;  
  
 if (potenciaDeMil % 2 == 1 && digitoEvaluado != 0) {  
 ordenDeMagnitudEnLetras = "Mil ";  
 }  
  
 switch (potenciaDeMil) {  
 case 2:  
 case 3:  
 return ordenDeMagnitudEnLetras + "Millones ";  
 case 4:  
 case 5:  
 return ordenDeMagnitudEnLetras + "Billones ";  
 case 6:  
 case 7:  
 return ordenDeMagnitudEnLetras + "Trillones ";  
  
 }  
  
 return ordenDeMagnitudEnLetras;  
 }  
  
 private static String convertirValorEntreCeroYCien(int digitoEvaluado) {  
 int unidades = digitoEvaluado % 10;  
 int decenas = digitoEvaluado / 10;  
  
 if (unidades == 0) {  
 return Decenas.convertir(decenas);  
 } else if (decenas == 0 || decenas >= 3) {  
 String y = "";  
 if (decenas != 0) {  
 y = "y ";  
 }  
 return Decenas.convertir(decenas) + y + Unidades.convertir(unidades);  
 } else {  
 if (digitoEvaluado <= 15) {  
 switch (unidades) {  
 case 1:  
 return "Once ";  
 case 2:  
 return "Doce ";  
 case 3:  
 return "Trece ";  
 case 4:  
 return "Catorce ";  
 case 5:  
 return "Quince ";  
 }  
 } else if (decenas == 1) {  
 return "Dieci" + Unidades.convertir(unidades).toLowerCase();  
 } else {  
 return "Veinti" + Unidades.convertir(unidades).toLowerCase();  
  
 }  
 }  
 return "";  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\utils\DateUtil.java

package mx.mbi.dgc.utils;  
  
  
public class DateUtil {  
 public DateUtil() {  
 super();  
 }  
 public static int diasDelMes(int mes, int año){  
 switch(mes){  
 case 0: // Enero  
 case 2: // Marzo  
 case 4: // Mayo  
 case 6: // Julio  
 case 7: // Agosto  
 case 9: // Octubre  
 case 11: // Diciembre  
 return 31;  
 case 3: // Abril  
 case 5: // Junio  
 case 8: // Septiembre  
 case 10: // Noviembre  
 return 30;  
 case 1: // Febrero  
 if ( ((año%100 == 0) && (año%400 == 0)) ||  
 ((año%100 != 0) && (año% 4 == 0)) )  
 return 29; // Año Bisiesto  
 else  
 return 28;  
 default:  
 throw new java.lang.IllegalArgumentException(  
 "El mes debe estar entre 0 y 11");  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\utils\DominioTool.java

package mx.mbi.dgc.utils;  
  
import java.math.BigDecimal;  
  
import java.util.List;  
import java.util.Map;  
import java.util.TreeMap;  
  
import mx.mbi.dgc.dao.DAOCatAnioMes;  
import mx.mbi.dgc.dao.DAOCatMesDia;  
import mx.mbi.dgc.dao.DAOCatPlanta;  
import mx.mbi.dgc.dao.negocio.DAOMesDiaFirma;  
import mx.mbi.dgc.dao.negocio.DaoNegocioCatCondiciones;  
import mx.mbi.dgc.dao.negocio.DaoNegocioCatFirmaCincominutal;  
import mx.mbi.dgc.jpa.entity.CatAnioMes;  
import mx.mbi.dgc.jpa.entity.CatCondiciones;  
import mx.mbi.dgc.jpa.entity.CatFirmaCincominutal;  
import mx.mbi.dgc.jpa.entity.CatMesDia;  
import mx.mbi.dgc.jpa.entity.CatMesDiaFirma;  
import mx.mbi.dgc.jpa.entity.CatPlanta;  
  
import org.joda.time.DateTime;  
import org.joda.time.format.DateTimeFormat;  
import org.joda.time.format.DateTimeFormatter;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class DominioTool {  
   
 @Autowired  
 DAOMesDiaFirma daoMesDiaFirma;  
   
 @Autowired  
 DAOCatPlanta daoCatPlanta;  
   
 @Autowired  
 DAOCatAnioMes daoCatAnioMes;  
   
 @Autowired  
 DAOCatMesDia daoCatMesDia;  
   
 @Autowired  
 DaoNegocioCatFirmaCincominutal daoCatFirmaCincominutal;  
   
 @Autowired  
 DaoNegocioCatCondiciones daoNegocioCatCondiciones;  
   
 public CatMesDiaFirma buildingCatMesDiaFirma(String formato, String cadenaTiempo, BigDecimal idPlanta){  
 CatMesDiaFirma catMesDiaFirma = null;  
 //dd/MM/yyyy hh:mm:ss.SSS a  
 cadenaTiempo = cadenaTiempo.replaceAll("p.m.", "pm");  
 cadenaTiempo = cadenaTiempo.replaceAll("a.m.", "am");  
 DateTimeFormatter builderDateTime = DateTimeFormat.forPattern(formato);  
   
 DateTimeFormatter builderDiaActual = DateTimeFormat.forPattern("dd");  
 DateTimeFormatter builderAnioMes = DateTimeFormat.forPattern("yyyy\_MM");  
 DateTimeFormatter builderFirma = DateTimeFormat.forPattern("HHmm");  
   
 DateTime firmaCinco = DateTime.parse(cadenaTiempo, builderDateTime);  
 DateTime firmaCincoMinus = firmaCinco.minusMinutes(5);  
   
 catMesDiaFirma = daoMesDiaFirma.getCatMesDiaFirma(  
 new BigDecimal(Integer.parseInt(builderDiaActual.print(firmaCinco))),  
 builderAnioMes.print(firmaCinco),   
 builderFirma.print(firmaCincoMinus) + builderFirma.print(firmaCinco),  
 idPlanta);  
 return catMesDiaFirma;  
 }  
   
 public CatMesDiaFirma getMinusCatMesDiaFirma(CatMesDiaFirma inicioTemporal, Integer unidadTiempo, Integer saltoTemporal){  
   
 CatMesDiaFirma retorno = null;  
   
 String yyyy\_MM = inicioTemporal.getCatMesDia().getCatAnioMes().getAnioMes();  
 String dd = inicioTemporal.getCatMesDia().getDia().toString();  
 //Se asegura de que el día tiene dos cifras  
 if(dd.length() < 2){  
 dd = "0" + dd;  
 }  
 String hhmm = inicioTemporal.getCatFirmaCincominutal().getFirma();  
 hhmm = hhmm.substring(4,hhmm.length());  
 String cadenaTiempo = yyyy\_MM + dd + hhmm;  
 DateTimeFormatter builderDateTimeFormat = null;  
 String pattern = "yyyy\_MMddHHmm";  
   
 builderDateTimeFormat = DateTimeFormat.forPattern(pattern);  
 DateTime inicioTemporalDateTime = DateTime.parse(cadenaTiempo, builderDateTimeFormat);  
 switch(unidadTiempo)  
 {  
 case 1://cincominutal  
 inicioTemporalDateTime = inicioTemporalDateTime.minusMinutes(saltoTemporal);  
 break;  
 case 2://mes  
 inicioTemporalDateTime = inicioTemporalDateTime.minusMonths(saltoTemporal);  
 break;  
 case 3://dia  
 inicioTemporalDateTime = inicioTemporalDateTime.minusDays(saltoTemporal);  
 break;  
 default:  
 }  
 retorno = buildingCatMesDiaFirma(  
 pattern,   
 builderDateTimeFormat.print(inicioTemporalDateTime),   
 inicioTemporal.getCatMesDia().getCatAnioMes().getCatPlanta().getPlantaId());  
 return retorno;  
 }  
   
 public Map<String, String> buildCondiciones(String agrupador, BigDecimal idPlanta){  
 CatPlanta catPlanta = daoCatPlanta.findById(idPlanta).orElse(new CatPlanta());  
 List<CatCondiciones> listaCondiciones = daoNegocioCatCondiciones.findAllByProcesoAndCatPlanta(agrupador, catPlanta);  
  
 Map<String, String> retorno = new TreeMap();  
 for(CatCondiciones condicion : listaCondiciones){  
 retorno.put(condicion.getKeyCondicion(), condicion.getValueCondicion());  
 }  
 return retorno;  
 }  
   
 public void generaFirmas(){  
 //America/Mexico\_City  
 DateTimeFormatter builderDateTimeFormat = DateTimeFormat.forPattern("yyyyMMddHHmm");  
 DateTimeFormatter fecha = DateTimeFormat.forPattern("yyyy/MM/dd");  
 DateTimeFormatter yyyy\_MM = DateTimeFormat.forPattern("yyyy\_MM");  
 String yyyy\_MM\_String ="-1";  
 String ddString = "-1";  
 DateTimeFormatter dd = DateTimeFormat.forPattern("dd");  
   
 DateTimeFormatter EEEEEEEEEE = DateTimeFormat.forPattern("EEEEEEEEEE");  
 DateTime inicioDateTime = DateTime.parse("200801010000", builderDateTimeFormat);  
 DateTime finDateTime = DateTime.parse("200801100000", builderDateTimeFormat);  
 CatAnioMes catAnioMes = null;  
 CatMesDia catMesDia = null;  
 CatMesDiaFirma catMesDiaFirma = null;  
 String yyyyMMString =yyyy\_MM.print(inicioDateTime);  
 List<CatPlanta> plantas = daoCatPlanta.findAll();  
 for(CatPlanta planta : plantas){  
 inicioDateTime = DateTime.parse("200801010000", builderDateTimeFormat);  
 do{  
   
 if(yyyy\_MM\_String.equals(yyyy\_MM.print(inicioDateTime))){  
 catAnioMes = new CatAnioMes();  
 catAnioMes.setCatPlanta(planta);  
 catAnioMes.setAnioMes(yyyy\_MM.print(inicioDateTime));  
 catAnioMes.setTotalDias(new BigDecimal(inicioDateTime.dayOfMonth().getMaximumValue()));  
 daoCatAnioMes.save(catAnioMes);  
 yyyy\_MM\_String = yyyy\_MM.print(inicioDateTime);  
 }  
   
 if(ddString.equals(yyyy\_MM.print(inicioDateTime))){  
 catMesDia = new CatMesDia();  
 catMesDia.setCatAnioMes(catAnioMes);  
 catMesDia.setDia(new BigDecimal(dd.print(inicioDateTime)));  
 catMesDia.setDiaSemana(EEEEEEEEEE.print(inicioDateTime));  
 catMesDia.setFechaCalendario(DateTime.parse(fecha.print(inicioDateTime), fecha).toDate());  
 daoCatMesDia.save(catMesDia);  
 ddString = dd.print(inicioDateTime);  
 }  
   
 catMesDiaFirma = new CatMesDiaFirma();  
 catMesDiaFirma.setCatMesDia(catMesDia);  
 catMesDiaFirma.setFechaFirma(DateTime.parse(fecha.print(inicioDateTime), fecha).toDate());  
 catMesDiaFirma.setCatFirmaCincominutal(getCatFirmaCincominutal(inicioDateTime));  
   
   
 inicioDateTime.plusMinutes(5);  
 }while(yyyyMMString.equals(yyyy\_MM.print(inicioDateTime)));  
 //catAnioMes.setTotalHoras(totalHoras);  
 }  
 }  
   
 private CatFirmaCincominutal getCatFirmaCincominutal(DateTime inicioDateTime){  
 CatFirmaCincominutal retorno = null;  
 retorno = daoCatFirmaCincominutal.getCatFirmaCincominutal("");  
 return retorno;  
 }  
   
 public boolean isUltimoDiaDelMes(CatMesDiaFirma inicioTemporal){  
 boolean retorno = false;  
 String pattern = "yyyy\_MMdd";  
 String cadenaTiempo =inicioTemporal.getCatMesDia().getCatAnioMes().getAnioMes() + inicioTemporal.getCatMesDia().getDia();  
 DateTimeFormatter builderDateTimeFormat = DateTimeFormat.forPattern(pattern);  
 DateTime inicioTemporalDateTime = DateTime.parse(cadenaTiempo, builderDateTimeFormat);  
 if(inicioTemporalDateTime.getDayOfMonth() == inicioTemporalDateTime.dayOfMonth().getMaximumValue()){  
 retorno = true;  
 }  
 return retorno;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\utils\exportadorexcel\ExportadorExcel.java

package mx.mbi.dgc.utils.exportadorexcel;  
  
import java.io.IOException;  
  
import java.math.BigDecimal;  
import java.math.BigInteger;  
import java.sql.ResultSet;  
import java.sql.ResultSetMetaData;  
import java.text.DateFormat;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
  
import java.util.\*;  
  
import mx.mbi.dgc.dao.negocio.DaoNegocioTxPerfilesVariable;  
import mx.mbi.dgc.jpa.entity.\*;  
  
import oracle.jdbc.OracleResultSetMetaData;  
import org.apache.poi.ss.usermodel.Row;  
import org.springframework.beans.factory.annotation.Autowired;  
  
  
/\*\* Maneja la exportacion a Excel para asi poder reutilizar el codigo mas facilmente para otras exportaciones.  
 \* Requiere de GeneradorExcel. \*/  
public class ExportadorExcel {  
 private static final GeneradorExcel.Estilos CABECERA = GeneradorExcel.Estilos.CABECERA;  
 private static final GeneradorExcel.Estilos BORDES = GeneradorExcel.Estilos.BORDES\_LIG;  
 private static final int COL\_NUM = 4; //Empieza en la 5a columna porque la primera columna es un indice y luego tres columnas para la fecha  
 private static final int ROW\_NUM = 15;  
 private static final BigDecimal diecisiete = new BigDecimal("17");  
 private static final BigDecimal dieciocho = new BigDecimal("18");  
 private static final DateFormat df = new SimpleDateFormat("dd/MM/yyyy");  
 private static final DateFormat fh = new SimpleDateFormat("dd/MM/yyyy H:mm");  
  
 @Autowired  
 private DaoNegocioTxPerfilesVariable daoNegocioTxPerfilesVariable;  
  
 private String tipo;  
 private String[] tipos;  
 private Integer valoresPorVariable; //Por defecto: valor original, formato, norma y dos estatus, deteccion y correccion  
 private GeneradorExcel generadorExcel;  
 private Row fila;  
 final int nFilaCabeceras = 14; //fila de cabeceras  
 private Integer rowNum = ROW\_NUM; //contador de filas existentes. Empieza en 15 porque arriba de eso va la cabecera.  
 private Integer colNum = COL\_NUM;  
 int nColEjeZ;  
 int variablesPorCincominutal = 1;  
 boolean bVariablesContadas = false;  
 private int filas = 0;  
 private Integer nIndefinidosCnt;  
  
 private String bufferFecha = " ";  
 private String bufferCincominutal = " ";  
  
 private String nombrePlanta;  
 private String fechaInicial;  
 private String fechaFinal;  
  
 //Para las columnas creadas  
 private HashMap<String,Integer> nombresColumnas = new HashMap<>();  
  
  
 /\*\* Se necesita inicializar con un nombre de Hoja y un nombre de Planta  
 \* @param nombreHoja nombre de la hoja a crear  
 \* @param nombrePlanta nombre de la planta para la que se realiza la exportacion  
 \* @param fechaInicial fecha inicial del rango de fechas a recuperar  
 \* @param fechaFinal fecha final del rango de fechas  
 \*/  
 public ExportadorExcel(String nombreHoja, String nombrePlanta, String fechaInicial, String fechaFinal){  
 generadorExcel = inicilizarExcel(nombreHoja);  
 fila = prepararExcel(generadorExcel, nombrePlanta, fechaInicial, fechaFinal);  
 }  
  
 /\*\* nueva version para exportar Formato, Norma y Perfiles juntos  
 \*  
 \* @param nombreHoja nombre de la hoja a crear  
 \* @param nombrePlanta nombre de la planta para la que se realiza la exportacion  
 \* @param fechaInicial fecha inicial del rango de fechas a recuperar  
 \* @param fechaFinal fecha final del rango de fechas  
 \* @param tipos tipos de datos a agregar  
 \*/  
 public ExportadorExcel(String nombreHoja, String nombrePlanta, String fechaInicial, String fechaFinal,  
 String[] tipos){  
 this.tipos = tipos;  
 valoresPorVariable = 3;  
 for (String tipo :  
 tipos) {  
 if (tipo.equals("Norma")||tipo.equals("Formato")) {  
 valoresPorVariable += 2;  
 }  
 }  
 System.out.println("columnas por variable: " + valoresPorVariable.toString());  
 generadorExcel = inicilizarExcel(nombreHoja);  
 this.nombrePlanta = nombrePlanta;  
 this.fechaInicial = fechaInicial;  
 this.fechaFinal = fechaFinal;  
 fila = prepararExcel(generadorExcel, nombrePlanta, fechaInicial, fechaFinal);  
 }  
  
 private ExportadorExcel(){}  
  
  
 /\*\* Genera y guarda un Excel en la direccion provista.  
 \* @param listaTx txCincominutalVar a ser insertados  
 \* @see mx.mbi.dgc.jpa.entity.TxCincominutalVar TxCincominutalVar  
 \* @param listaCat catalogo de nombres de estatus de la variable  
 \* @see mx.mbi.dgc.jpa.entity.CatEstatusVariable  
 \* @param filePath donde se va a guardar el archivo  
 \* @param tipo el tipo de exportacion (Formato, Norma)  
 \* @return si se pudo realizar correctamente\*/  
 public Boolean generarExcel(List<TxCincominutalVar> listaTx, List<CatEstatusVariable> listaCat,  
 String filePath,  
 String tipo) {  
 nIndefinidosCnt = 0;  
// int entradasEscritas = 0;  
  
 //Iteracion de base  
 for (TxCincominutalVar entry : listaTx) {  
// System.out.printf("Escribiendo variable %s\nID:", entry.getCatVariable().getNombre());  
// System.out.println(entry.getCatVariable().getVariableId());  
  
 if (entry.getCatVariable().getVariableId().equals(diecisiete)||entry.getCatVariable().getVariableId().equals(dieciocho)){  
// System.out.printf("Saltando la variable : %s", entry.getCatVariable().getNombre());  
 continue;  
 }  
  
/\* System.out.printf("Entradas escritas: %d\n", entradasEscritas);  
 if (entradasEscritas>=listaTx.size()-variablesPorCincominutal-1){  
 break;  
 }  
  
 entradasEscritas++;\*/  
  
 revisarFechaHora(entry);  
  
 //Verifica si existe una columna para esa variable y la ubica o la crea inserta el nombre de variable.  
 String sNombreVariable = entry.getCatVariable().getNombre();  
 String sNombreUnidad = entry.getCatVariable().getUnidadMedida();  
 int nColActual = insertarColumnasSiNoExisten(entry, valoresPorVariable, sNombreVariable, sNombreUnidad, tipo);  
 nColEjeZ = 0;  
  
 //Se escriben los valores de la variable en las columnas correspondientes  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ++, entry.getValorOriginal(), true);  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ++, entry.getValorFormato(), false);  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ++, entry.getValorNorma(), false);  
  
 //Se escriben los [nombre] de [Cat\_Estatus\_Variable] de deteccion y correcion de norma y formato en lugar de las ID  
 if(tipo.equals("Formato")){  
 escribirEstatusEnCelda(listaCat, nColActual, entry.getEstatusDf(), nColEjeZ++);  
 escribirEstatusEnCelda(listaCat, nColActual, entry.getEstatusCf(), nColEjeZ++);  
 }  
 if(tipo.equals("Norma")){  
 escribirEstatusEnCelda(listaCat, nColActual,entry.getEstatusDn(), nColEjeZ++);  
 escribirEstatusEnCelda(listaCat, nColActual,entry.getEstatusCn(), nColEjeZ);  
 }  
  
 }  
  
 fila = generadorExcel.elegirFila(1);  
 generadorExcel.crearCelda(fila, 1, filas);  
  
 return escribir(filePath);  
 }  
  
 /\*\*genera un excel con el orden dado por tags  
 \*  
 \* @param listaTx  
 \* @param listaCat  
 \* @param tags  
 \* @param filePath  
 \* @param tipo  
 \* @return  
 \*/  
 public boolean generarExcel(List<TxCincominutalVar> listaTx, List<CatEstatusVariable> listaCat, List<CatTags> tags,  
 String filePath,  
 String tipo) {  
 this.tipo = tipo;  
 prepararOrdenDeColumnas(listaTx, tags);  
 return generarExcel(listaTx, listaCat, filePath, tipo);  
 }  
  
 public void generarExcel(ResultSet resultSet, String filePath) {  
 try {  
 OracleResultSetMetaData resultSetMetaData = (OracleResultSetMetaData) resultSet.getMetaData();  
  
 //Leemos las columnas de la tabla  
 if (resultSetMetaData == null||resultSet==null) {  
 System.out.println("No hay datos o metadatos disponibles");  
 } else {  
 crearHeaders(resultSetMetaData);  
 int nColumnas = resultSetMetaData.getColumnCount();  
 System.out.printf("Columnas encontradas: %d\n", nColumnas);  
 while (resultSet.next()) {  
 fila = generadorExcel.elegirFila(rowNum++);  
 for (int i = 1; i <= nColumnas - 5; i++) {  
 generadorExcel.crearCelda(fila, i, resultSet.getString(i), false);  
 }  
 generadorExcel.crearCelda(fila, nColumnas - 4, resultSet.getInt(nColumnas - 4), false);  
 generadorExcel.crearCelda(fila, nColumnas - 3, resultSet.getDate(nColumnas - 3), false);  
 generadorExcel.crearCelda(fila, nColumnas - 2, resultSet.getInt(nColumnas - 2), false);  
 generadorExcel.crearCelda(fila, nColumnas - 1, resultSet.getInt(nColumnas - 1), false);  
 generadorExcel.crearCelda(fila, nColumnas, resultSet.getDate(nColumnas), false);  
 }  
 }  
  
  
 escribir(filePath);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 }  
  
 @SuppressWarnings("Duplicates")  
 /\*\*genera un excel ordenado y con posiblemente más de un tipo  
 \*  
 \* @param listaTx  
 \* @param listaEstatus  
 \* @param tags  
 \* @param filePath  
 \* @param tipos  
 \* @return  
 \*/  
 public boolean generarExcel(List<TxCincominutalVar> listaTx, List<CatEstatusVariable> listaEstatus,  
 List<CatTags> tags,  
 String filePath,  
 String[] tipos){  
 this.tipos = tipos;  
 prepararOrdenDeColumnas(listaTx,tags);  
 nIndefinidosCnt = 0;  
// int entradasEscritas = 0;  
  
  
 System.out.printf("Tamaño de la lista: %d\n",listaTx.size());  
 for (TxCincominutalVar entry : listaTx) {  
  
 BigDecimal varID = entry.getCatVariable().getVariableId();  
  
 if (varID.equals(diecisiete)||varID.equals(dieciocho)){  
 continue;  
 }  
  
 revisarFechaHora(entry);  
  
 //Verifica si existe una columna para esa variable y la ubica o la crea inserta el nombre de variable.  
 String sNombreVariable = entry.getCatVariable().getNombre();  
 String sNombreUnidad = entry.getCatVariable().getUnidadMedida();  
  
 int nColActual;  
 nColActual = insertarColumnasSiNoExisten(entry, valoresPorVariable, sNombreVariable, sNombreUnidad, tipos);  
  
 int nColEjeZ = 0;  
  
 boolean bFormatoNorma = false;  
 for (String tipo :  
 tipos) {  
 if(tipo.equals("Formato")||tipo.equals("Norma")){  
 bFormatoNorma = true;  
 }  
 }  
  
 if(bFormatoNorma){  
  
 //Se escriben los valores de la variable en las columnas correspondientes  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ++, entry.getValorOriginal(), true);  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ++, entry.getValorFormato(), false);  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ++, entry.getValorNorma(), false);  
 }  
  
 for (String tipo : tipos) {  
 //Se escriben los [nombre] de [Cat\_Estatus\_Variable] de deteccion y correcion de norma y formato  
 if(tipo.equals("Formato")){  
 escribirEstatusEnCelda(listaEstatus, nColActual, entry.getEstatusDf(), nColEjeZ++);  
 escribirEstatusEnCelda(listaEstatus, nColActual, entry.getEstatusCf(), nColEjeZ++);  
 }  
 if(tipo.equals("Norma")){  
 escribirEstatusEnCelda(listaEstatus, nColActual,entry.getEstatusDn(), nColEjeZ++);  
 escribirEstatusEnCelda(listaEstatus, nColActual,entry.getEstatusCn(), nColEjeZ++);  
 }  
 }  
 }  
 return escribir(filePath);  
 }  
  
 @SuppressWarnings("Duplicates")  
 /\*\*genera un excel ordenado y con posiblemente más de un tipo  
 \*  
 \* @param listaTx  
 \* @param listaEstatus  
 \* @param tags  
 \* @param filePath  
 \* @param tipos  
 \* @return  
 \*/  
 public boolean generarExcel(List<TxCincominutalVar> listaTx, List<CatEstatusVariable> listaEstatus,  
 List<CatTags> tags,  
 String filePath,  
 String[] tipos, String ultimoDia){  
 this.tipos = tipos;  
 prepararOrdenDeColumnas(listaTx,tags);  
 nIndefinidosCnt = 0;  
// int entradasEscritas = 0;  
  
  
 System.out.printf("Tamaño de la lista: %d\n",listaTx.size());  
 for (TxCincominutalVar entry : listaTx) {  
  
 BigDecimal varID = entry.getCatVariable().getVariableId();  
  
 if (varID.equals(diecisiete)||varID.equals(dieciocho)){  
 continue;  
 }  
  
 if (!revisarFechaHora(entry, true)) {  
 Date fechaVerificacion = entry.getTxCincominutal().getCatMesDiaFirma().getFechaFirma();  
 SimpleDateFormat dia = new SimpleDateFormat("dd/MMM/yyyy");  
  
 try {  
 SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MMM/yyyy H:mm");  
 ultimoDia = ultimoDia.toLowerCase();  
// Date fechaBuffer = dateFormat.parse(ultimoDia);  
// ultimoDia = fh.format(fechaBuffer);  
 SimpleDateFormat hora = new SimpleDateFormat("H:mm");  
 String sDiaV = dia.format(fechaVerificacion);  
 String sHoraV = hora.format(fechaVerificacion);  
 String sFecha = sDiaV + " " + sHoraV;  
// System.out.println("Fecha del cincominutal: " + sFecha);  
// System.out.println("Dia: " + sDiaV + " - Hora: " + sHoraV);  
// System.out.println("Fecha de Ultimo Dia: " + ultimoDia);  
 if(ultimoDia.trim().equals(sFecha.trim())){  
 System.out.println("Saltando el ultimo cincominutal");  
 break;  
 }  
  
 } catch (Exception e){  
 e.printStackTrace();  
 }  
  
 }  
 revisarFechaHora(entry);  
  
 //Verifica si existe una columna para esa variable y la ubica o la crea inserta el nombre de variable.  
 String sNombreVariable = entry.getCatVariable().getNombre();  
 String sNombreUnidad = entry.getCatVariable().getUnidadMedida();  
  
 int nColActual;  
 nColActual = insertarColumnasSiNoExisten(entry, valoresPorVariable, sNombreVariable, sNombreUnidad, tipos);  
  
 int nColEjeZ = 0;  
  
 boolean bFormatoNorma = false;  
 for (String tipo :  
 tipos) {  
 if(tipo.equals("Formato")||tipo.equals("Norma")){  
 bFormatoNorma = true;  
 }  
 }  
  
 if(bFormatoNorma){  
  
 //Se escriben los valores de la variable en las columnas correspondientes  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ++, entry.getValorOriginal(), true);  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ++, entry.getValorFormato(), false);  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ++, entry.getValorNorma(), false);  
 }  
  
 for (String tipo : tipos) {  
 //Se escriben los [nombre] de [Cat\_Estatus\_Variable] de deteccion y correcion de norma y formato  
 if(tipo.equals("Formato")){  
 escribirEstatusEnCelda(listaEstatus, nColActual, entry.getEstatusDf(), nColEjeZ++);  
 escribirEstatusEnCelda(listaEstatus, nColActual, entry.getEstatusCf(), nColEjeZ++);  
 }  
 if(tipo.equals("Norma")){  
 escribirEstatusEnCelda(listaEstatus, nColActual,entry.getEstatusDn(), nColEjeZ++);  
 escribirEstatusEnCelda(listaEstatus, nColActual,entry.getEstatusCn(), nColEjeZ++);  
 }  
 }  
 }  
 return escribir(filePath);  
 }  
  
 @SuppressWarnings("Duplicates")  
 public boolean generarExcel(List<TxCincominutalVar> listaCincominutalVars, List<CatEstatusVariable> listaEstatus,  
 List<CatTags> tags, String[] tipos,  
 List<TxPerfilesVariable> listaPerfiles, List<CatEstatusPerfil> listaCatPerfiles,  
 String filePath, BigDecimal plantaID, String ultimoDia){  
 this.tipos = tipos;  
 prepararOrdenDeColumnas(listaCincominutalVars,tags);  
// int entradasEscritas = 0;  
  
  
 //Iteracion de base  
 for (TxCincominutalVar entry : listaCincominutalVars) {  
 BigDecimal varID = entry.getCatVariable().getVariableId();  
  
 if (varID.equals(diecisiete)||varID.equals(dieciocho)){  
 continue;  
 }  
/\* if (entradasEscritas>=listaCincominutalVars.size()-variablesPorCincominutal-1){  
 break;  
 }  
 entradasEscritas++;\*/  
  
 revisarFechaHora(entry);  
  
 //Verifica si existe una columna para esa variable y la ubica o la crea inserta el nombre de variable.  
 String sNombreVariable = entry.getCatVariable().getNombre();  
 String sNombreUnidad = entry.getCatVariable().getUnidadMedida();  
  
 int nColActual;  
 nColActual = insertarColumnasSiNoExisten(entry, valoresPorVariable, sNombreVariable, sNombreUnidad, tipos);  
  
 int nColEjeZ = 0;  
  
 //Se escriben los valores de la variable en las columnas correspondientes  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ++, entry.getValorOriginal(), true);  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ++, entry.getValorFormato(), false);  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ++, entry.getValorNorma(), false);  
  
  
 for (String tipo : tipos) {  
 //Se escriben los [nombre] de [Cat\_Estatus\_Variable] de deteccion y correcion de norma y formato  
 if(tipo.equals("Formato")){  
 escribirEstatusEnCelda(listaEstatus, nColActual, entry.getEstatusDf(), nColEjeZ++);  
 escribirEstatusEnCelda(listaEstatus, nColActual, entry.getEstatusCf(), nColEjeZ++);  
 }  
 if(tipo.equals("Norma")){  
 escribirEstatusEnCelda(listaEstatus, nColActual,entry.getEstatusDn(), nColEjeZ++);  
 escribirEstatusEnCelda(listaEstatus, nColActual,entry.getEstatusCn(), nColEjeZ++);  
 }  
 }  
 }  
  
 // escribir(filePath);  
  
 generadorExcel.crearHoja("Perfiles");  
  
 rowNum= ROW\_NUM;  
 filas = 0;  
  
 fila = prepararExcel(generadorExcel, nombrePlanta, fechaInicial, fechaFinal);  
  
 return generarExcelPerfiles(listaPerfiles,listaCatPerfiles,filePath, ultimoDia);  
 }  
  
  
  
 @SuppressWarnings("Duplicates")  
 /\*\* igual que el anterior pero para Perfiles. No necesita definir un tipo.  
 \* @param listaTx txCincominutalVar a ser insertados  
 \* @see mx.mbi.dgc.jpa.entity.TxCincominutalVar TxCincominutalVar  
 \* @param listaCat catalogo de nombres de estatus de la variable  
 \* @see mx.mbi.dgc.jpa.entity.CatEstatusVariable  
 \* @param filePath donde se va a guardar el archivo  
 \* @return si se pudo realizar correctamente\*/  
 public Boolean generarExcelPerfiles(List<TxPerfilesVariable> listaTx, List<CatEstatusPerfil> listaCat,  
 String filePath, String ultimoDia){  
 nIndefinidosCnt = 0;  
// int entradasEscritas = 0;  
  
 //Iteracion de base  
 for (TxPerfilesVariable entry : listaTx) {  
 if (!revisarFechaHora(entry, true)) {  
 Date fechaVerificacion = entry.getTxCincominutal().getCatMesDiaFirma().getFechaFirma();  
 SimpleDateFormat dia = new SimpleDateFormat("dd/MMM/yyyy");  
  
 try {  
 SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MMM/yyyy H:mm");  
 ultimoDia = ultimoDia.toLowerCase();  
// Date fechaBuffer = dateFormat.parse(ultimoDia);  
// ultimoDia = fh.format(fechaBuffer);  
 SimpleDateFormat hora = new SimpleDateFormat("H:mm");  
 String sDiaV = dia.format(fechaVerificacion);  
 String sHoraV = hora.format(fechaVerificacion);  
 String sFecha = sDiaV + " " + sHoraV;  
// System.out.println("Fecha del cincominutal: " + sFecha);  
// System.out.println("Dia: " + sDiaV + " - Hora: " + sHoraV);  
// System.out.println("Fecha de Ultimo Dia: " + ultimoDia);  
 if(ultimoDia.trim().equals(sFecha.trim())){  
 System.out.println("Saltando el ultimo cincominutal");  
 break;  
 }  
  
 } catch (Exception e){  
 e.printStackTrace();  
 }  
  
 }  
  
 revisarFechaHora(entry);  
  
 //Verifica si existe una columna para esa variable y la ubica o la crea inserta el nombre de variable.  
 String sNombreVariable;  
 String sNombreUnidad;  
 if(entry.getCatVariablePerfil()!=null){  
 //System.out.println(entry.getCatVariablePerfil().getVariablePerfil());  
 sNombreVariable = entry.getCatVariablePerfil().getVariablePerfil();  
 sNombreUnidad = entry.getCatVariablePerfil().getUnidadPerfil();  
 } else {  
 nIndefinidosCnt ++;  
 //System.out.println("No hay un Variable\_Perfil\_Id definido");  
 sNombreVariable = "(indefinido: " + nIndefinidosCnt.toString() + ")";  
 sNombreUnidad = "(indefinido)";  
 }  
  
 int nColActual = insertarColumnasSiNoExisten(entry, 3, sNombreVariable, sNombreUnidad,  
 "Perfiles");  
  
 //Se escriben los valores de la variable en las columnas correspondientes  
 generadorExcel.crearCelda(fila, nColActual, entry.getValor(),true);  
 String sNombreMedidor = null;  
 if (entry.getCatMedidores()!=null) {  
 sNombreMedidor = entry.getCatMedidores().getNombre();  
 } else {  
 sNombreMedidor = "(indefinido)";  
 }  
 generadorExcel.crearCelda(fila, nColActual+1, sNombreMedidor,false);  
  
 String sNombreEstatus;  
 if (entry.getCatEstatusPerfil()!=null) {  
 sNombreEstatus = entry.getCatEstatusPerfil().getEstatusPerfil();  
 } else {  
 sNombreEstatus = "(indefinido)";  
 }  
// generadorExcel.crearCelda(fila, nColActual+2, sNombreEstatus,false);  
 escribirEstatusPerfilEnCelda(listaCat,nColActual,entry.getCatEstatusPerfil().getEstatusPerfilId(),2);  
 }  
  
 fila = generadorExcel.elegirFila(1);  
 generadorExcel.crearCelda(fila, 1, filas);  
  
 return escribir(filePath);  
 } //FIN: generarExcel()  
  
 private static GeneradorExcel inicilizarExcel(String nombreHoja) {  
 //Creamos un cuaderno, una hoja y un estilo  
 GeneradorExcel generadorExcel = new GeneradorExcel();  
 generadorExcel.crearHoja(nombreHoja);  
 return generadorExcel;  
 } //FIN: inicializarExcel()  
  
 private Row prepararExcel(GeneradorExcel generadorExcel, String nombrePlanta) {  
 Row fila;  
 //Nombre de la planta  
 fila = generadorExcel.elegirFila(0);  
 generadorExcel.crearCelda(fila, 0, nombrePlanta,false);  
 generadorExcel.unirCeldas(0,0,0,4); //Une las celdas del nombre de la planta  
  
 fila = generadorExcel.elegirFila(1);  
 generadorExcel.crearCelda(fila, 0, "Registros: ");  
  
 //Creamos la hilera de los encabezados y agregamos el de Fecha y Hora  
 fila = generadorExcel.elegirFila(14);  
 generadorExcel.crearCelda(fila, 0, "Registro",CABECERA);  
 generadorExcel.crearCelda(fila, 1, "Fecha y Hora Inicio",CABECERA);  
 generadorExcel.crearCelda(fila, 2,"Fecha y Hora Fin",CABECERA);  
 generadorExcel.crearCelda(fila, 3,"Verificacion",CABECERA);  
 return fila;  
 }  
  
 private Row prepararExcel(GeneradorExcel generadorExcel, String nombrePlanta, String fechaInicial, String fechaFinal) {  
 prepararExcel(generadorExcel, nombrePlanta);  
 Row fila = generadorExcel.elegirFila(4);  
  
 crearHeaderColspanDos(fila, generadorExcel, 4, 9, "Periodo");  
 fila = generadorExcel.elegirFila(5);  
 generadorExcel.crearCelda(fila,9,fechaInicial,BORDES);  
 generadorExcel.crearCelda(fila,10,fechaFinal,BORDES);  
  
 fila = generadorExcel.elegirFila(4);  
 generadorExcel.crearCelda(fila,11,"Rango",CABECERA);  
 fila = generadorExcel.elegirFila(5);  
 generadorExcel.crearCelda(fila,11,"Cincominutal",BORDES);  
 return fila;  
 }  
  
 //Usa la primera fila de datos para ordenar las variables y luego crear las columnas correspondientes en Excel  
 private void prepararOrdenDeColumnas(List<TxCincominutalVar> listaDeVariables, List<CatTags> tags){  
 //En la primera iteracion sFechaHoraCm es necesariamente diferente de fechaBuffer  
 Boolean bPrimeraIteracion = true;  
 String fechaBuffer="";  
 TreeMap<BigInteger, TxCincominutalVar> listaOrdenada = new TreeMap<>();  
  
 for (TxCincominutalVar variable : listaDeVariables) {  
 //Revisa si fue la ultima variable del cincominutal  
 Date fechaHoraCm = variable.getTxCincominutal().getCatMesDiaFirma().getFechaFirma();  
 DateFormat fh = new SimpleDateFormat("dd/MM/yyyy HH:mm");  
 String sFechaHoraCm = fh.format(fechaHoraCm);  
  
 //Continua si la hora del cincominutal no ha cambiado, en caso contrario termina el bucle  
 if (!(sFechaHoraCm.equals(fechaBuffer)||bPrimeraIteracion)) { break; }  
  
 BigInteger posicion = obtenerPosicionDeOrdenamiento(variable, tags);  
 if (posicion.equals(BigInteger.ZERO)){  
 System.out.println("No hay un orden definido para " + variable.getCatVariable().getNombre() + ". Omitiendo.");  
 } else {  
 listaOrdenada.put(posicion, variable);  
 }  
  
 fechaBuffer = sFechaHoraCm;  
 bPrimeraIteracion = false;   
  
 }  
  
 for (TxCincominutalVar variableOrdenada : listaOrdenada.values()){  
 String sNombre = variableOrdenada.getCatVariable().getNombre();  
 String sUnidad = variableOrdenada.getCatVariable().getUnidadMedida();  
  
 insertarColumnasSiNoExisten(variableOrdenada, valoresPorVariable, sNombre, sUnidad, tipos);  
 }  
  
 }  
  
  
 private void crearHeaders(ResultSetMetaData resultSetMetaData) {  
 try {  
 int nColumnas = resultSetMetaData.getColumnCount();  
 fila = generadorExcel.elegirFila(nFilaCabeceras);  
 for (int i = 1; i <= nColumnas; i++) {  
// System.out.println(i);  
  
 generadorExcel.crearCelda(fila, i, resultSetMetaData.getColumnName(i),  
 CABECERA);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 }  
  
 private void revisarFechaHora(IFTxCincominutalable entry) { //Verifica si entramos a un nuevo cincominutal  
 //Los cincominutales estan ordenados por fecha y hora, solo necesita checar si es igual al anterior  
 Date fechaHoraCm = entry.getTxCincominutal().getCatMesDiaFirma().getFechaFirma();  
 String sFechaHoraCm = fh.format(fechaHoraCm);  
  
 //Como String.substring es destructivo, se necesitan crear buffers para formatear correctamente las horas de inicio y fin  
 String fecha = df.format(fechaHoraCm);  
 String sHora = entry.getTxCincominutal().getCatMesDiaFirma().getCatFirmaCincominutal().getFirma();  
 String horaFin = sHora.substring(4);  
 String horaInicio = sHora.substring(0, 4);  
 String minutos = horaInicio.substring(2);  
 String horas = horaInicio.substring(0,2);  
 sHora = horas + ":" + minutos;  
 horaInicio = sHora;  
 minutos = horaFin.substring(2);  
 horas = horaFin.substring(0,2);  
 sHora = horas + ":" + minutos;  
 horaFin = sHora;  
  
 //Si no es igual, crea una nueva hilera (fila) e inyecta la fecha y hora  
 if (!sFechaHoraCm.equals(bufferFecha)) {  
 crearNuevaFilaCincominutal(fechaHoraCm, df, fecha, horaInicio, horaFin);  
// System.out.printf("Variables por cincominutal: %d\n", variablesPorCincominutal);  
 if(variablesPorCincominutal != 0){  
 bVariablesContadas = true;  
 }  
 } else if (!bVariablesContadas){  
 variablesPorCincominutal++;  
 }  
  
 bufferFecha=sFechaHoraCm;  
 }  
  
 private Boolean revisarFechaHora(IFTxCincominutalable entry, Boolean b) {  
 if (!b){revisarFechaHora(entry);}  
  
 //Los cincominutales estan ordenados por fecha y hora, solo necesita checar si es igual al anterior  
 Date fechaHoraCm = entry.getTxCincominutal().getCatMesDiaFirma().getFechaFirma();  
 DateFormat fh = new SimpleDateFormat("dd/MM/yyyy HH:mm");  
 String sFechaHoraCm = fh.format(fechaHoraCm);  
  
  
 return sFechaHoraCm.equals(bufferFecha);  
  
 }  
 private void crearNuevaFilaCincominutal(Date fechaHoraCm, DateFormat df, String fecha,  
 String horaInicio, String horaFin) {  
 nIndefinidosCnt = 0; //Resetea la cuenta de columnas indefinidas  
 fila = generadorExcel.elegirFila(rowNum++);  
 colNum = COL\_NUM;  
  
  
 generadorExcel.crearCelda(fila, 0, ++filas,false);  
 generadorExcel.crearCelda(fila, 1, fecha + " " + horaInicio,false);  
 //Pasamos al dia siguiente si la hora final es 00:00  
 if (horaFin.equals("00:00")){  
 Calendar calendario = Calendar.getInstance();  
 calendario.setTime(fechaHoraCm);  
 calendario.add(Calendar.DATE, 1);  
 fecha = df.format(calendario.getTime());  
  
 }  
 generadorExcel.crearCelda(fila, 2, fecha + " " + horaFin,false);  
  
 //Creamos la columna de verificacion  
 String sFormula;  
 if(rowNum<=16){  
 sFormula = "0";  
 generadorExcel.crearCelda(fila, 3, sFormula,false,true);  
 } else {  
 Integer rowNumPrevio = rowNum - 1;  
 sFormula = "B" + rowNum.toString() + "-C" + rowNumPrevio.toString();  
 generadorExcel.crearCelda(fila, 3, sFormula,false, true);  
 }  
 }  
  
  
  
 private int insertarColumnasSiNoExisten(Object entry, int nColumnas, String sNombreVariable, String sNombreUnidad,  
 String tipo) {  
  
 boolean bExiste = false;  
 int nColActual = 0; //Para evitar el error mas adelante porque el compilador no entiende que siempre se le asigna un valor.  
 //Verificamos si existe la Variable que buscamos  
 for (String s : nombresColumnas.keySet()) {  
 if(sNombreVariable.equals(s)){  
 nColActual = nombresColumnas.get(s);  
 bExiste = true;  
 break;  
 }  
 }  
  
 //Si la variable no existe, se crean las columnas correspondientes.  
 if(!bExiste){  
  
 //Selecciona o elige la fila a trabajar.  
  
 Row filaPrimera = (generadorExcel.elegirFila(nFilaCabeceras -1)!=null) ? generadorExcel.elegirFila(nFilaCabeceras -1):generadorExcel.elegirFila(nFilaCabeceras -1);  
 Row fila = (generadorExcel.elegirFila(nFilaCabeceras)!=null) ? generadorExcel.elegirFila(nFilaCabeceras):generadorExcel.elegirFila(nFilaCabeceras);  
  
 nColActual = colNum;  
 colNum = colNum + nColumnas; //Se usan nColumnas columnas para deteccion y correccion por variable  
  
  
 if (entry instanceof TxCincominutalVar) {  
 generadorExcel.crearCelda(fila,nColActual,sNombreVariable + " (" + sNombreUnidad + ")",CABECERA);  
 generadorExcel.crearCelda(fila,nColActual+1,"Valor de Formato",  
 CABECERA);  
 generadorExcel.crearCelda(fila,nColActual+2,  
 "Valor Corregido",CABECERA);  
  
  
 if (tipo.equals("Formato")||tipo.equals("Norma")){  
 crearHeaderColspanDos(filaPrimera, generadorExcel, nFilaCabeceras -1, nColActual+3, tipo);  
 generadorExcel.crearCelda(fila,nColActual+3,"Deteccion",  
 CABECERA);  
 generadorExcel.crearCelda(fila,nColActual+4,"Correccion",  
 CABECERA);  
  
 }  
  
 nombresColumnas.put(sNombreVariable,nColActual);  
 } else if(entry instanceof TxPerfilesVariable) {  
 generadorExcel.crearCelda(fila, nColActual,  
 sNombreVariable + " (" + sNombreUnidad + ")",CABECERA);  
 generadorExcel.crearCelda(fila,nColActual+1,"Medidor",  
 CABECERA);  
 generadorExcel.crearCelda(fila,nColActual+2,"Estatus",  
 CABECERA);  
 nombresColumnas.put(sNombreVariable,nColActual);  
 }  
  
 }  
 return nColActual;  
 }  
  
 private int insertarColumnasSiNoExisten(Object entry, int nColumnas, String sNombreVariable, String sNombreUnidad,  
 String[] tipos) {  
  
 boolean bExiste = false;  
 int nColActual = 0; //Para evitar el error mas adelante porque el compilador no entiende que siempre se le asigna un valor.  
 //Verificamos si existe la Variable que buscamos  
 for (String s : nombresColumnas.keySet()) {  
 if(sNombreVariable.equals(s)){  
 nColActual = nombresColumnas.get(s);  
 bExiste = true;  
 break;  
 }  
 }  
  
 //Si la variable no existe, se crean las columnas correspondientes.  
 if(!bExiste){  
  
 //Selecciona o elige la fila a trabajar.  
  
 Row filaPrimera = (generadorExcel.elegirFila(nFilaCabeceras -1)!=null) ? generadorExcel.elegirFila(nFilaCabeceras -1):generadorExcel.elegirFila(nFilaCabeceras -1);  
 Row fila = (generadorExcel.elegirFila(nFilaCabeceras)!=null) ? generadorExcel.elegirFila(nFilaCabeceras):generadorExcel.elegirFila(nFilaCabeceras);  
  
 nColActual = colNum;  
 colNum = colNum + valoresPorVariable; //Se usan nColumnas columnas para deteccion y correccion por variable  
  
  
 if (entry instanceof TxCincominutalVar) {  
 generadorExcel.crearCelda(fila,nColActual,sNombreVariable + " (" + sNombreUnidad + ")",CABECERA);  
 generadorExcel.crearCelda(fila,nColActual+1,"Valor de Formato", CABECERA);  
 generadorExcel.crearCelda(fila,nColActual+2,"Valor Corregido",CABECERA);  
  
  
 int nColEjeZ = 3 ;  
 if (tipos[0].equals("Formato")&&tipos.length>1&&tipos[1].equals("Norma")) {  
 crearHeaderColspanDos(filaPrimera, generadorExcel, nFilaCabeceras - 1, nColActual + nColEjeZ, tipos[0]);  
 nColEjeZ +=2;  
 crearHeaderColspanDos(filaPrimera, generadorExcel, nFilaCabeceras - 1, nColActual + nColEjeZ, tipos[1]);  
  
 nColEjeZ = 3;  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ++, "Deteccion",  
 CABECERA);  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ++, "Correccion",  
 CABECERA);  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ++, "Deteccion",  
 CABECERA);  
 generadorExcel.crearCelda(fila, nColActual + nColEjeZ, "Correccion",  
 CABECERA);  
  
 } else if (tipos[0].equals("Formato")||tipos[0].equals("Norma")){  
 crearHeaderColspanDos(filaPrimera, generadorExcel, nFilaCabeceras -1, nColActual+3, tipos[0]);  
 generadorExcel.crearCelda(fila,nColActual+3,"Deteccion",  
 CABECERA);  
 generadorExcel.crearCelda(fila,nColActual+4,"Correccion",  
 CABECERA);  
  
 }  
  
 nombresColumnas.put(sNombreVariable,nColActual);  
 } else if(entry instanceof TxPerfilesVariable) {  
 generadorExcel.crearCelda(fila, nColActual,  
 sNombreVariable + " (" + sNombreUnidad + ")",CABECERA);  
 generadorExcel.crearCelda(fila,nColActual+1,"Medidor",  
 CABECERA);  
 generadorExcel.crearCelda(fila,nColActual+2,"Estatus",  
 CABECERA);  
 nombresColumnas.put(sNombreVariable,nColActual);  
 }  
  
 }  
 return nColActual;  
 }  
  
  
 private BigInteger obtenerPosicionDeOrdenamiento(TxCincominutalVar variable, List<CatTags> tags){  
 BigInteger resultado = BigInteger.ZERO;  
 boolean bEncontrado = false;  
 Integer tamano = tags.size();  
 //System.out.println("Cantidad de tags:" + tamano.toString());  
 for (CatTags tag : tags) {  
 //System.out.println("Agregando Tag");  
  
 if (tag.getVariableId()!=null) {  
 //System.out.println(tag.getVariableId());  
 if ((tag.getOrden()!=null)){  
 //System.out.println("Orden encontrado " + tag.getOrden().toString());  
 }  
 if ((tag.getVariableId().equals(variable.getCatVariable().getVariableId().toBigInteger()))){  
 //System.out.println("Escribiendo " + variable.getCatVariable().getNombre());  
 resultado = tag.getOrden();  
 }  
 }  
 }  
  
 return resultado;  
 }  
  
 private String encontrarNombreEstatus(List<CatEstatusVariable> lista, BigDecimal Id){  
 for (CatEstatusVariable estatus: lista){  
 if(estatus.getEstatusVarId().equals(Id)){  
 return estatus.getNombre();  
 }  
 }  
  
 return "(indefinido)"; //Si no encuentra un estatus con el Id buscado  
 }  
  
 private String encontrarColorEstatus(List<CatEstatusVariable> lista, BigDecimal Id){  
 for (CatEstatusVariable estatus: lista){  
 if (estatus.getColor()!=null){  
 if(estatus.getEstatusVarId().equals(Id)){  
 return estatus.getColor();  
 }  
 }  
 }  
  
 return null; //Si no encuentra un estatus con el Id buscado  
 }  
  
  
 private Boolean escribir(String filePath) {  
 try {  
 System.out.println("Guardando Excel en " + filePath);  
 generadorExcel.finalizarEscritura(filePath);  
 return true;  
 } catch (IOException ioe) {  
 ioe.printStackTrace();  
 return false;  
 }  
 }  
  
  
 private void escribirEstatusEnCelda(List<CatEstatusVariable> listaCat, int nColActual, BigDecimal estatus, int i) {  
  
 String nombreEstatus = encontrarNombreEstatus(listaCat, estatus);  
 generadorExcel.crearCelda(fila, nColActual + i, nombreEstatus, false);  
  
 String estiloEstatus = encontrarColorEstatus(listaCat,estatus);  
 if (estiloEstatus!=null){  
 generadorExcel.estilizarCeldas(fila.getRowNum(),nColActual + i, estiloEstatus);  
 }  
  
 }  
  
 private String encontrarColorEstatusPerfil(List<CatEstatusPerfil> lista, BigDecimal Id){  
 for (CatEstatusPerfil estatus: lista){  
 if (estatus.getColor()!=null){  
 if(estatus.getEstatusPerfilId().equals(Id)){  
 return estatus.getColor();  
 }  
 }  
 }  
 return null; //Si no encuentra un estatus con el Id buscado  
 }  
  
 private String encontrarNombreEstatusPerfil(List<CatEstatusPerfil> lista, BigDecimal Id){  
 for (CatEstatusPerfil estatus: lista){  
 if(estatus.getEstatusPerfilId().equals(Id)){  
 return estatus.getEstatusPerfil();  
 }  
 }  
 return "(indefinido)"; //Si no encuentra un estatus con el Id buscado  
 }  
  
 private void escribirEstatusPerfilEnCelda(List<CatEstatusPerfil> listaCat, int nColActual, BigDecimal estatus, int i) {  
  
 String nombreEstatus = encontrarNombreEstatusPerfil(listaCat, estatus);  
 generadorExcel.crearCelda(fila, nColActual + i, nombreEstatus, false);  
  
 String estiloEstatus = encontrarColorEstatusPerfil(listaCat,estatus);  
 if (estiloEstatus!=null){  
 generadorExcel.estilizarCeldas(fila.getRowNum(),nColActual + i, estiloEstatus);  
 }  
  
 }  
  
 private void crearHeaderColspanDos(Row filaPrimera, GeneradorExcel generadorExcel, int i, int i2, String tipo) {  
 generadorExcel.unirCeldas(i, 0, i2, 1);  
 generadorExcel.estilizarCeldas(i, 0, i2, 1, CABECERA);  
 generadorExcel.crearCelda(filaPrimera, i2, tipo, false);  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\utils\exportadorexcel\GeneradorExcel.java

package mx.mbi.dgc.utils.exportadorexcel;  
  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
  
import java.io.IOException;  
  
import java.math.BigDecimal;  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.Iterator;  
  
import org.apache.poi.xssf.usermodel.IndexedColorMap;  
import org.apache.poi.xssf.usermodel.XSSFCellStyle;  
import org.apache.poi.xssf.usermodel.XSSFColor;  
import org.apache.poi.xssf.usermodel.XSSFWorkbook;  
import org.apache.poi.ss.usermodel.\*;  
import org.apache.poi.ss.util.CellRangeAddress;  
  
/\*\* Esta clase contiene servicio para facilitar el manejo y generacion de archivos Excel. Su constructor es privado  
 \* porque esta clase no se instancia. \*/  
public class GeneradorExcel{  
 public enum Estilos{  
 CABECERA(0),  
 BORDE\_IZQ(1),  
 BORDE\_DER(2),  
 BORDE\_ARR(3),  
 BORDE\_ABA(4),  
 BORDES\_LIG(5),  
 NARANJA(6),  
 AZUL(7),  
 AMARILLO(8),  
 ROJO(9),  
 VERDE(10);  
  
  
 private int nIndice;  
  
 Estilos(int n){  
 this.nIndice = n;  
 }  
  
 int Indice (){return nIndice;}  
  
 }  
 private Workbook cuaderno;  
  
 private Sheet hoja;  
 private static final int tamano = Estilos.values().length + 1;  
 private CellStyle[] estilos = new CellStyle[tamano];  
 private CellStyle estiloBuffer = null;  
 private HashMap<String, CellStyle> colores = new HashMap<>();  
 private FileInputStream fileInputStream;  
  
 /\*\* constructor. Genera un nuevo cuaderno y los estilos de base. \*/  
 public GeneradorExcel(){  
 cuaderno = new XSSFWorkbook();  
 inicializarEstilos();  
 }  
  
 public GeneradorExcel(String plantilla){  
 try {  
 fileInputStream = new FileInputStream(new File(plantilla));  
 cuaderno = WorkbookFactory.create(fileInputStream);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
  
  
 /\*\* Crea una nueva hoja con un nombre nombreHoja.  
 \* @param nombreHoja <code>nombre</code> de la hoja  
 \* @return una <code>hoja</code> Excel en la memoria\*/  
 public Sheet crearHoja(String nombreHoja) {  
 System.out.println("Creando hoja " + nombreHoja);  
 hoja = cuaderno.createSheet(nombreHoja);  
 return hoja;  
 }  
  
  
 /\*\* La version mas simple de crearFila()  
 \* @param rowNum <code>numero</code> de fila (empieza en 0)  
 \* @return referencia a la <code>fila</code> creada\*/  
 public Row crearFila(int rowNum) {  
 Row fila = hoja.createRow(rowNum);  
 return fila;  
 }  
  
  
 /\*\* Sobrecarga crearFila para crear una <code>fila</code> ya poblada con cualquier cantidad de objetos  
 \* que pueden ser de tipos diferentes (por eso usa <code>varargs</code> y no un <code>array</code>).  
 \* No devuelve nada porque la idea es que sea una sola llamada para crear toda la <code>fila</code> y que no sea  
 \* necesario crear celdas.  
 \* @param hoja <code>hoja</code> en la que se va a crear  
 \* @param rowNum <code>numero</code> de fila (empieza en 0)  
 \* @param columnas lista de objetos a insertar (separados por una coma). Pueden ser de tipos diferentes.\*/  
 public void crearFila(Sheet hoja, int rowNum, Object... columnas) {  
 int nCnt = 0;  
 Row fila = hoja.createRow(rowNum);  
 for (Object columna : columnas) {  
 crearCelda(fila,nCnt,columna,false);  
 nCnt++;  
 }  
 }  
  
 /\*\* Elige una fila en la hoja actual  
 \* @param nFila el numero de fila a elegir (empieza en 0)  
 \* @return retorna una fila para poder modificarla\*/  
 public Row elegirFila(int nFila){  
 Row fila = (hoja.getRow(nFila)!=null) ? hoja.getRow(nFila) : hoja.createRow(nFila);  
 return fila;  
 }  
  
  
 /\*\* metodo polimorfico para crear una <code>celda</code> independientemente del tipo de objeto  
 \* @param fila <code>fila</code> en la que se va a crear (vertical de arriba hacia abajo). Empieza en cero.  
 \* @param nColumna <code>numero</code> de celda dentro de la fila (horizontal de izquierda a derecha). Empieza en cero.  
 \* @param valor Puede ser String, Int, Double o Float.  
 \* @param bBorde Se usa cuando se quiere agregar un borde del lado izquierdo. \*/  
 public Cell crearCelda(Row fila, int nColumna, Object valor, Boolean bBorde) {  
 Cell celda;  
 celda = (fila.getCell(nColumna)!=null) ? fila.getCell(nColumna) : fila.createCell(nColumna);  
  
 if (valor instanceof String) {  
 celda.setCellValue((String) valor);  
 } else if (valor instanceof Integer){  
 celda.setCellValue((Integer) valor);  
 } else if (valor instanceof Double){  
 celda.setCellValue((Double) valor);  
 } else if (valor instanceof Float){  
 celda.setCellValue((Float) valor);  
 } else if (valor instanceof Date){  
 celda.setCellValue((Date) valor);  
 CreationHelper ch = cuaderno.getCreationHelper();  
 estiloBuffer.setDataFormat(ch.createDataFormat().getFormat("dd/MM/yyyy h:mm"));  
 celda.setCellStyle(estiloBuffer);  
 } else if (valor!=null){  
 celda.setCellValue(valor.toString());  
 } else {  
 celda.setCellValue("(vacio)");  
 }  
  
 if(bBorde&&(estiloBuffer !=null)){  
 celda.setCellStyle(estiloBuffer);  
 }  
 return celda;  
 }  
  
 /\*\*Crea una celda con un estilo aplicado.  
 \*  
 \* @param fila fila en donde va a ubicarse  
 \* @param nColumna numero de columna  
 \* @param valor valor a introducir en la celda  
 \* @param estilos estilos a aplicar  
 \* @return retorna una celda  
 \*/  
 public Cell crearCelda(Row fila, int nColumna, Object valor, Estilos... estilos){  
 Cell celda = crearCelda(fila,nColumna,valor,false);  
 for (Estilos estilo: estilos) {  
 celda.setCellStyle(this.estilos[estilo.Indice()]);  
 }  
  
 return celda;  
 }  
  
 /\*\* Sobrecarga crearCelda() con una bandera para que inserte una formula y no un Objeto  
 \* \*/  
 public void crearCelda(Row fila, int nColumna, Object valor, Boolean bBorde, Boolean esFormula){  
  
 Cell celda = fila.createCell(nColumna);  
 celda.setCellType(CellType.FORMULA);  
 celda.setCellFormula(valor.toString());  
 }  
  
 /\*\* para unir celdas en una sola celda mas grande  
 \*  
 \* @param nRowInicial  
 \* @param nRows  
 \* @param nColInicial  
 \* @param nCols  
 \*/  
 public void unirCeldas(int nRowInicial, int nRows, int nColInicial, int nCols){  
 hoja.addMergedRegion(new CellRangeAddress(nRowInicial,nRowInicial + nRows,  
 nColInicial,nColInicial + nCols));  
 }  
  
 /\*\*Aplica el mismo estilo a una o mas celdas  
 \*  
 \* @param filaInicial numero de fila inicial. Empieza en 0.  
 \* @param filas cantidad de filas a seleccionar ademas de la primera (elegir 0 para una sola fila)  
 \* @param colInicial numero de columna inicial. Empieza en 0.  
 \* @param cols cantidad de columnas a seleccionar ademas de la primera  
 \* @param estilo estilo a aplicar  
 \*/  
 public void estilizarCeldas(int filaInicial, int filas, int colInicial, int cols, Estilos estilo) {  
 //Aplicamos el estilo a cada columna de cada fila de la region seleccionada  
 CellStyle estiloAplicado = this.estilos[estilo.Indice()];  
 for (int y = filaInicial; y <= filaInicial + filas; y++) {  
 Row fila = elegirFila(y); //Selecciona la fila si existe, si no, la crea  
 for (int x = colInicial; x <= colInicial + cols; x++) {  
 //Si existe, selecciona la celda, si no, la crea  
 if (fila.getCell(x) != null) {  
 fila.getCell(x).setCellStyle(estiloAplicado);  
 } else {  
 fila.createCell(x).setCellStyle(estiloAplicado);  
 }  
  
 }  
 }  
 }  
  
 /\*\*Aplica un color RGB en una celda  
 \* @param fila numero de fila inicial. Empieza en 0.  
 \* @param columna numero de columna inicial. Empieza en 0.  
 \* @param estilo color a aplicar en formato RRR,GGG,BBB  
 \*/  
 public void estilizarCeldas(int fila, int columna, String estilo) {  
 if (estilo == null) {  
 estilo = ""; // para evitar un null pointer exception mas adelante  
 }  
 Boolean existe = false;  
 CellStyle cellStyle = null;  
  
 for (String color: colores.keySet()) {  
 if (estilo.equals(color)){  
 existe = true;  
 cellStyle = colores.get(color);  
 break;  
 }  
 }  
 //Si existe el color, lo selecciona, si no, lo crea y lo anade al HashMap de estilos de color  
 if (!existe){  
 cellStyle = cuaderno.createCellStyle();  
 String[] sRgb = estilo.trim().split(",");  
 if (sRgb.length==3) {  
 if (!(sRgb[0].trim().equals(""))&&!(sRgb[1].trim().equals(""))&&!(sRgb[2].trim().equals(""))) {  
 short[] rgb = new short[sRgb.length];  
 int nCont = 0;  
 for (String valor: sRgb) {  
 rgb[nCont++] = Short.parseShort(valor.trim());  
 // System.out.printf("%d, %d, %d\n",rgb[0],rgb[1],rgb[2]);  
 }  
 IndexedColorMap colorMap = null;  
 XSSFColor color = new XSSFColor(new java.awt.Color(rgb[0],rgb[1],rgb[2]));  
 ((XSSFCellStyle) cellStyle).setFillForegroundColor(color);  
 cellStyle.setFillPattern(FillPatternType.SOLID\_FOREGROUND);  
 colores.put(estilo,cellStyle);  
 } else {  
 System.out.println("Color mal definido");  
 }  
 } else {  
 System.out.println("El color no tiene los parametros correctos");  
 }  
 }  
 Cell celda = (elegirFila(fila).getCell(columna)!=null) ? elegirFila(fila).getCell(columna) : elegirFila(fila).createCell(columna);  
 celda.setCellStyle(cellStyle);  
 }  
  
 /\*\*  
 \* Busca los placeholders definidos dentro de (la primera hoja de) un archivo Excel.  
 \* @param simboloInicial el símbolo usado para marcar el placeholder al inicio  
 \* @param simboloFinal el símbolo usado para marcar el placeholder al final. Para un excel es redundante pero puede ser útil en el caso de un Word.  
 \* @return las coordenadas agrupadas por placeholder  
 \*/  
 public HashMap<String, ArrayList<Integer[]>> escanearExcel(String simboloInicial, String simboloFinal) {  
 //TODO: ¿Refactorizar para no necesitar símbolo final?  
 HashMap<String, ArrayList<Integer[]>> resultado = new HashMap<>();  
 System.out.printf("Simbolo inicial: %s\nSimbolo final: %s\n", simboloInicial, simboloFinal);  
 try {  
 Sheet hoja = cuaderno.getSheetAt(0);  
 int nFila = 0;  
 for (Row fila : hoja) {  
 int nCelda = 0;  
 for (Cell celda : fila) {  
 if (celda.getCellTypeEnum().equals(CellType.STRING)) {  
 String sCelda = celda.getStringCellValue();  
 String placeholder = celda.getStringCellValue();  
 if (sCelda.length() > simboloInicial.length()) {  
 System.out.printf("Contenido de la celda %d - %d: %s\n", nFila, nCelda, sCelda);  
 String inicioDeCelda = sCelda.substring(0,simboloInicial.length());  
 System.out.printf("Simbolos iniciales: %s\n", inicioDeCelda);  
 if (simboloInicial.equals(inicioDeCelda)) {  
 if (sCelda.length() > simboloFinal.length()) {  
 String finDeCelda = sCelda.substring(sCelda.length() - simboloFinal.length());  
 System.out.printf("Simbolos finales: %s\n", finDeCelda);  
 if (simboloFinal.equals(finDeCelda)) {  
 Integer[] coordenadas = {nFila, nCelda};  
 resultado.computeIfAbsent(placeholder, k -> new ArrayList<>()); //Si no existe la entrada en el hashmap, la crea  
 resultado.get(placeholder).add(coordenadas);  
 System.out.printf("Encontrado %s en %d - %d\n", sCelda, coordenadas[0], coordenadas[1]);  
 }  
 }  
  
 }  
 }  
 }  
  
 nCelda++;  
 }  
 nFila++;  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return resultado;  
 }  
  
 /\*\*  
 \* Actualiza una celda en la hoja 0.  
 \* @param coordenadasFilaColumna las coordenadas de la celda, fila primero, columna segundo  
 \* @param valor el valor a introducir.  
 \*/  
 public void actualizarCelda(Integer[] coordenadasFilaColumna, Object valor) {  
 Sheet hoja = cuaderno.getSheetAt(0);  
 Row fila = hoja.getRow(coordenadasFilaColumna[0]);  
 Cell celda = fila.getCell(coordenadasFilaColumna[1]);  
 if (valor instanceof BigDecimal) {  
 BigDecimal valorInsertado = (BigDecimal) valor;  
 celda.setCellType(CellType.NUMERIC);  
 celda.setCellValue(valorInsertado.doubleValue());  
 } else if (valor instanceof String) {  
 String valorInsertado = (String) valor;  
 celda.setCellType(CellType.STRING);  
 celda.setCellValue(valorInsertado);  
 }  
  
 }  
  
 /\*\* Escribe el <code>cuaderno</code> elegido en el archivo de nombre especificado y lo cierra para hacer definitivos  
 \* los cambios.  
 \* @param ubicacion <code>direccion</code> en donde se va guardar. Debe incluir el nombre del archivo (.xlsx).  
 \* @throws IOException Sucede si hay una excepcion de Entrada/Salida\*/  
 public void finalizarEscritura(String ubicacion) throws IOException{  
 FileOutputStream salida = new FileOutputStream(ubicacion);  
 cuaderno.setForceFormulaRecalculation(true);  
 cuaderno.write(salida);  
 cuaderno.close();  
 salida.close();  
 }  
  
 public void cerrarLector() throws IOException{  
 fileInputStream.close();  
 }  
  
 private void inicializarEstilos() {  
 //Inicializamos los estilos  
 int nCount = 0;  
 for (CellStyle cs: estilos){  
 estilos[nCount++] = cuaderno.createCellStyle();  
 }  
 crearEstilo(Estilos.CABECERA);  
 crearEstilo(Estilos.BORDE\_IZQ);  
 crearEstilo(Estilos.BORDES\_LIG);  
 crearEstilo(Estilos.AMARILLO);  
 crearEstilo(Estilos.NARANJA);  
 crearEstilo(Estilos.AZUL);  
 crearEstilo(Estilos.ROJO);  
 crearEstilo(Estilos.VERDE);  
  
 crearEstiloBordeIzq();  
 }  
  
 private void crearEstilo(Estilos estilo){  
 int indice = estilo.Indice();  
 switch (estilo){  
 case CABECERA:  
 estilos[indice].setFillForegroundColor(IndexedColors.GREY\_40\_PERCENT.getIndex());  
 estilos[indice].setFillPattern(FillPatternType.SOLID\_FOREGROUND);  
 estilos[indice].setBorderTop(BorderStyle.THIN);  
 estilos[indice].setBorderRight(BorderStyle.THIN);  
 estilos[indice].setBorderBottom(BorderStyle.THIN);  
 estilos[indice].setBorderLeft(BorderStyle.THIN);  
 estilos[indice].setAlignment(HorizontalAlignment.CENTER);  
 break;  
 case BORDE\_IZQ:  
 estilos[indice].setBorderLeft(BorderStyle.MEDIUM);  
 break;  
 case BORDES\_LIG:  
 estilos[indice].setBorderTop(BorderStyle.THIN);  
 estilos[indice].setBorderRight(BorderStyle.THIN);  
 estilos[indice].setBorderBottom(BorderStyle.THIN);  
 estilos[indice].setBorderLeft(BorderStyle.THIN);  
 estilos[indice].setAlignment(HorizontalAlignment.CENTER);  
 break;  
 case AZUL:  
 estilos[indice].setFillForegroundColor(IndexedColors.AQUA.getIndex());  
 estilos[indice].setFillPattern(FillPatternType.SOLID\_FOREGROUND);  
 break;  
 case ROJO:  
 estilos[indice].setFillForegroundColor(IndexedColors.RED.getIndex());  
 estilos[indice].setFillPattern(FillPatternType.SOLID\_FOREGROUND);  
 break;  
 case VERDE:  
 estilos[indice].setFillForegroundColor(IndexedColors.LIGHT\_GREEN.getIndex());  
 estilos[indice].setFillPattern(FillPatternType.SOLID\_FOREGROUND);  
 break;  
 case NARANJA:  
 estilos[indice].setFillForegroundColor(IndexedColors.ORANGE.getIndex());  
 estilos[indice].setFillPattern(FillPatternType.SOLID\_FOREGROUND);  
 break;  
 case AMARILLO:  
 estilos[indice].setFillForegroundColor(IndexedColors.YELLOW.getIndex());  
 estilos[indice].setFillPattern(FillPatternType.SOLID\_FOREGROUND);  
 break;  
 }  
 }  
  
 private void crearEstiloBordeIzq(){  
 System.out.println("Creando estiloBuffer de celdas...");  
 estiloBuffer = cuaderno.createCellStyle();  
 estiloBuffer.setBorderLeft(BorderStyle.THICK);  
 }  
  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\utils\exportadorexcel\package-info.java

/\*\* paquete exportador de datos a Excel\*/  
package mx.mbi.dgc.utils.exportadorexcel;

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\utils\JRUtil.java

package mx.mbi.dgc.utils;  
  
import java.lang.reflect.Method;  
  
import java.math.BigDecimal;  
  
import java.util.Calendar;  
import java.util.Date;  
  
import javax.script.ScriptEngine;  
import javax.script.ScriptEngineManager;  
  
public class JRUtil {  
 static ScriptEngineManager manager = new ScriptEngineManager();  
 static ScriptEngine engine = manager.getEngineByName("js");  
  
 /\*\*  
 \* Este método copia todas las propiedades en cascada que sean identicas en nombre y tipo de retorno de un bean a un entity o viceversa  
 \* @param to  
 \* @param from  
 \*/  
 public static void copyProperties(Object to, Object from) {  
 Method[] metodos = from.getClass().getMethods();  
 Class<?>[] types = new Class[] { };  
 Object[] args = new Object[] { };  
 for (Method metodo : metodos) {  
 String name = metodo.getName();  
 Method metodoGetTo = null;  
 Method metodoSetTo = null;  
 Method metodoGetFrom = null;  
 Object objFrom = null;  
 Object objAux = null;  
 String get = null;  
 if (name.startsWith("set")) {  
 try {  
 get = name.replaceFirst("set", "get");  
 metodoGetTo = to.getClass().getMethod(get, types);  
 metodoSetTo = to.getClass().getMethod(name, metodoGetTo.getReturnType());  
 metodoGetFrom = from.getClass().getMethod(get, types);  
 objFrom = metodoGetFrom.invoke(from, args);  
 metodoSetTo.invoke(to, objFrom);  
 } catch (NoSuchMethodException nsme) {  
 } catch (IllegalArgumentException ilegal) {  
 try {  
 objAux = metodoGetTo.getReturnType().newInstance();  
 copyProperties(objAux, objFrom);  
 metodoSetTo.invoke(to, objAux);  
 } catch (Exception ex) {  
 }  
 } catch (Exception e) {  
 }  
 }  
 }  
 }  
  
 public static String getAnioMes() {  
 Calendar fecha = Calendar.getInstance();  
 int anio = fecha.get(Calendar.YEAR);  
 int mes = fecha.get(Calendar.MONTH) + 1;  
 String anioMes = anio + "\_" + (mes < 10 ? "0" + mes : mes);  
 //System.out.println("anioMes: " + anioMes);  
 return anioMes;  
 }  
  
 public static String getAnioMesAnterior() {  
 Calendar fecha = Calendar.getInstance();  
 int anio = fecha.get(Calendar.YEAR);  
 int mes = fecha.get(Calendar.MONTH) + 1;  
 if (mes == 1) {  
 anio = anio - 1;  
 mes = 12;  
 } else {  
 mes = mes - 1;  
 }  
 String anioMes = anio + "\_" + (mes < 10 ? "0" + mes : mes);  
 //System.out.println("anioMes anterior: " + anioMes);  
 return anioMes;  
 }  
  
 public static String getDiaActual() {  
 Calendar fecha = Calendar.getInstance();  
 int dia = fecha.get(Calendar.DAY\_OF\_MONTH);  
 String diaActual = dia < 10 ? "0" + dia : "" + dia;  
 //System.out.println("getDiaActual: " + diaActual);  
 return diaActual;  
 }  
  
 public static int getDiaDelMes() {  
 Calendar fecha = Calendar.getInstance();  
 return fecha.get(Calendar.DAY\_OF\_MONTH);  
 }  
  
 public static int getUltimoDiaMesAnterior() {  
 Calendar fecha = Calendar.getInstance();  
 int anio = fecha.get(Calendar.YEAR);  
 int mes = fecha.get(Calendar.MONTH) - 1;  
 int dia = fecha.get(Calendar.DAY\_OF\_MONTH);  
 fecha.set(anio, mes, dia);  
 if (mes <= 0 || mes == 1) {  
 anio = fecha.get(Calendar.YEAR) - 1;  
 mes = 12;  
 dia = 1;  
 fecha.set(anio, mes, dia);  
 }  
 int ultimoDia = fecha.getActualMaximum(Calendar.DAY\_OF\_MONTH);  
 System.out.println("Último día del mes anterior: " + ultimoDia + " - mes: " + (mes + 1));  
 return ultimoDia;  
 }  
  
 public static Date getFechaUltimoDiaMesAnterior() {  
 Calendar fecha = Calendar.getInstance();  
 int anio = fecha.get(Calendar.YEAR);  
 int mes = fecha.get(Calendar.MONTH) - 1;  
 int dia = fecha.get(Calendar.DAY\_OF\_MONTH);  
 fecha.set(anio, mes, dia);  
 if (mes <= 0 || mes == 1) {  
 anio = fecha.get(Calendar.YEAR) - 1;  
 mes = 12;  
 dia = 1;  
 fecha.set(anio, mes, dia);  
 }  
 int ultimoDia = fecha.getActualMaximum(Calendar.DAY\_OF\_MONTH);  
 fecha.set(anio, mes, ultimoDia);  
 System.out.println("#### Fecha Último día del mes anterior: " + fecha.getTime());  
 return fecha.getTime();  
 }  
  
 public static Date getUltimoDiaMesActual() {  
 Calendar fecha = Calendar.getInstance();  
 int anio = fecha.get(Calendar.YEAR);  
 int mes = fecha.get(Calendar.MONTH) + 1;  
 int dia = fecha.get(Calendar.DAY\_OF\_MONTH);  
 int ultimoDia = fecha.getActualMaximum(dia);  
 Calendar cal = Calendar.getInstance();  
 cal.set(anio, mes, ultimoDia);  
 System.out.println("#### Último día del mes actual: " + cal.getTime());  
 return cal.getTime();  
 }  
  
 public static double interpolacion(int x, int n, BigDecimal a, BigDecimal b) throws Exception {  
 engine.put("x", x); //x = 1 hasta n = número de errores  
 engine.put("a", a);  
 engine.put("b", b);  
 engine.put("n", n);  
 Object interpolLineal = engine.eval("x \*((b - a) / (n + 1)) + a"); //¿Por qué n+1?  
 return (Double) interpolLineal;  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\utils\XML.java

package mx.mbi.dgc.utils;  
  
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;  
  
import org.w3c.dom.DOMImplementation;  
import org.w3c.dom.Document;  
import org.w3c.dom.Element;  
import org.w3c.dom.Text;  
  
public class XML {  
   
 public static void main(String[] args) {  
 Document xml = null;  
 try {  
 DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();  
 DocumentBuilder builder;  
 builder = factory.newDocumentBuilder();  
 DOMImplementation implementation = builder.getDOMImplementation();  
 xml = implementation.createDocument(null, "cfdiXml", null);  
 xml.setXmlVersion("1.0");  
 Element comprobante = xml.createElement("cfdi:Comprobante");  
 comprobante.setAttribute("xmlns:cfdi", "http://www.sat.gob.mx/cfd/3");  
 comprobante.setAttribute("xmlns:xsi", " http://www.w3.org/2001/XMLSchema-instance");  
 comprobante.setAttribute("xsi:schemaLocation",  
 "http://www.sat.gob.mx/cfd/3 http://www.sat.gob.mx/sitio\_internet/cfd/3/cfdv33.xsd");  
 comprobante.setAttribute("Version", "3.3");  
  
 //Creación de elementos  
 //creamos el elemento principal casa  
 Element casa = xml.createElement("Casa");   
 //creamos un nuevo elemento. Casa contiene habitaciones  
 Element habitacion= xml.createElement("Habitacion");  
 //creamos un nuevo elemento. Habitación tiene color  
 Element color = xml.createElement("Color");   
 //Ingresamos la info. El color de esta habitación es azul  
 Text valorColor = xml.createTextNode("Azul");   
  
 //Asignamos la versión de nuestro XML  
 xml.setXmlVersion("1.0");   
 //Añadimos la casa al documento  
 xml.getDocumentElement().appendChild(casa);   
 //Añadimos el elemento hijo a la raíz  
 casa.appendChild(habitacion);   
 //Añadimos elemento  
 habitacion.appendChild(color);   
 //Añadimos valor  
 color.appendChild(valorColor);   
 }catch(Exception e){  
 System.err.println("Error");  
 }  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\java\mx\mbi\dgc\utils\zipeador\Zipeador.java

package mx.mbi.dgc.utils.zipeador;  
  
import java.io.File;  
import java.io.FileInputStream;  
  
import java.io.FileOutputStream;  
import java.util.ArrayList;  
import java.util.zip.ZipEntry;  
import java.util.zip.ZipOutputStream;  
  
public class Zipeador {  
 public static void generarZip(String ubicacion, ArrayList<String> documentosAComprimir) {  
 try {  
 FileOutputStream salidaDeDatos = new FileOutputStream(ubicacion);  
 ZipOutputStream salidaZip = new ZipOutputStream(salidaDeDatos);  
 for (String documento : documentosAComprimir) {  
 File archivoAZipear = new File(documento);  
 FileInputStream lectorDeArchivo = new FileInputStream(archivoAZipear);  
 ZipEntry entradaAlZip = new ZipEntry(archivoAZipear.getName());  
 salidaZip.putNextEntry(entradaAlZip);  
  
 byte[] bytes = new byte[1024];  
 int tamano;  
 while ((tamano = lectorDeArchivo.read(bytes)) >= 0) {  
 salidaZip.write(bytes, 0, tamano);  
 }  
 lectorDeArchivo.close();  
 }  
 salidaZip.close();  
 salidaDeDatos.close();  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
  
 }  
}

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\resources\application-dev.properties

#server.port=9080  
logging.level.org.springframework.web=INFO  
logging.level.mx.mbi.dgc=INFO  
# logs the SQL statements  
log4j.logger.org.hibernate.SQL=debug   
# Logs the JDBC parameters passed to a query  
log4j.logger.org.hibernate.type=trace   
hibernate.show\_sql=true  
logging.file=/Proyectos/Test/logs/DGC/safe/safeMSA.log  
spring.datasource.url=jdbc:oracle:thin:@200.52.85.139:1521/devicollab.dgc.com  
spring.datasource.username=QADGC  
spring.datasource.password=rxVLet4P  
spring.datasource.driver.class=oracle.jdbc.driver.OracleDriver  
spring.jpa.show-sql=false  
spring.jpa.properties.hibernate.default\_schema=QADGC  
spring.jpa.properties.hibernate.hbm2ddl.auto=none  
#spring.jpa.properties.hibernate.hbm2ddl.auto=validate  
#spring.jpa.properties.hibernate.hbm2ddl.auto=create  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.Oracle12cDialect  
#spring.datasource.jndi-name=jdbc/safeDS  
  
#Spring MultiPartFile Configuration  
spring.servlet.multipart.enabled=true  
# Threshold after which files are written to disk.  
#spring.servlet.multipart.file-size-threshold=2KB  
# Max file size.  
spring.servlet.multipart.max-file-size=200MB  
# Max Request Size  
spring.servlet.multipart.max-request-size=215MB  
  
utils.exportadorexcel.path=/etc/nginx/html/dgc/excel/

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\resources\application-qa.properties

logging.level.org.springframework.web=INFO  
logging.level.mx.mbi.dgc=INFO  
log4j.logger.org.hibernate.SQL=debug   
# Logs the JDBC parameters passed to a query  
log4j.logger.org.hibernate.type=trace   
hibernate.show\_sql=true  
logging.file=/Proyectos/Test\\logs\\DGC\\safe\\safeMSA.log  
#logging.file=/Programas/dev/logs/safeMSA.log  
  
spring.datasource.url=jdbc:oracle:thin:@200.52.85.139:1521/icollabpdb.dgc.com  
#spring.datasource.url=jdbc:oracle:thin:@172.20.141.101:1521/icollabpdb.dgc.com  
spring.datasource.username=QADGC  
spring.datasource.password=wJ5nZAUW  
spring.datasource.driver.class=oracle.jdbc.driver.OracleDriver  
spring.jpa.show-sql=false  
spring.jpa.properties.hibernate.default\_schema=QADGC  
spring.jpa.properties.hibernate.hbm2ddl.auto=none  
#spring.jpa.properties.hibernate.hbm2ddl.auto=validate  
#spring.jpa.properties.hibernate.hbm2ddl.auto=create  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.Oracle12cDialect  
#spring.datasource.jndi-name=jdbc/safeDS  
  
#Spring MultiPartFile Configuration  
spring.servlet.multipart.enabled=true  
# Threshold after which files are written to disk.  
#spring.servlet.multipart.file-size-threshold=2KB  
# Max file size.  
spring.servlet.multipart.max-file-size=200MB  
# Max Request Size  
spring.servlet.multipart.max-request-size=215MB  
  
utils.exportadorexcel.path=/Programas/qa/excel

C:\Users\guadalupe.lopez\Desktop\SAFE\MSA\src\main\resources\application.properties

spring.profiles.active=@activatedProperties@