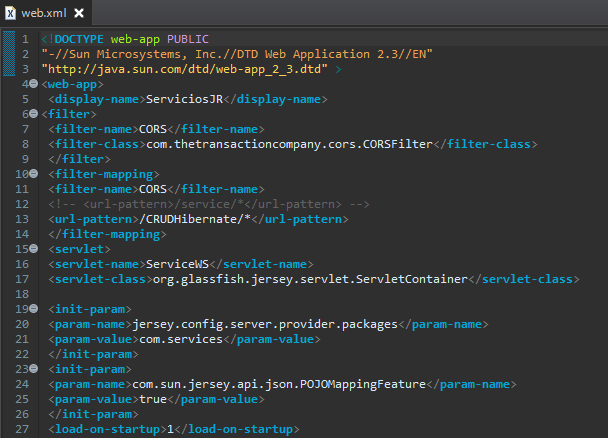
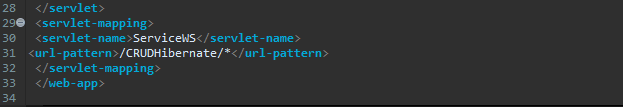
Nombre del proyecto: **CRUDHibernate**



Web.xml





<!**DOCTYPE** web-app PUBLIC

"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"

"http://java.sun.com/dtd/web-app\_2\_3.dtd" >

<**web-app**>

<**display-name**>ServiciosJR</**display-name**>

<**filter**>

<**filter-name**>CORS</**filter-name**>

<**filter-class**>com.thetransactioncompany.cors.CORSFilter</**filter-class**>

</**filter**>

<**filter-mapping**>

<**filter-name**>CORS</**filter-name**>

<!-- <url-pattern>/service/\*</url-pattern> -->

<**url-pattern**>/CRUDHibernate/\*</**url-pattern**>

</**filter-mapping**>

<**servlet**>

<**servlet-name**>ServiceWS</**servlet-name**>

<**servlet-class**>org.glassfish.jersey.servlet.ServletContainer</**servlet-class**>

<**init-param**>

<**param-name**>jersey.config.server.provider.packages</**param-name**>

<**param-value**>com.services</**param-value**>

</**init-param**>

<**init-param**>

<**param-name**>com.sun.jersey.api.json.POJOMappingFeature</**param-name**>

<**param-value**>true</**param-value**>

</**init-param**>

<**load-on-startup**>1</**load-on-startup**>

</**servlet**>

<**servlet-mapping**>

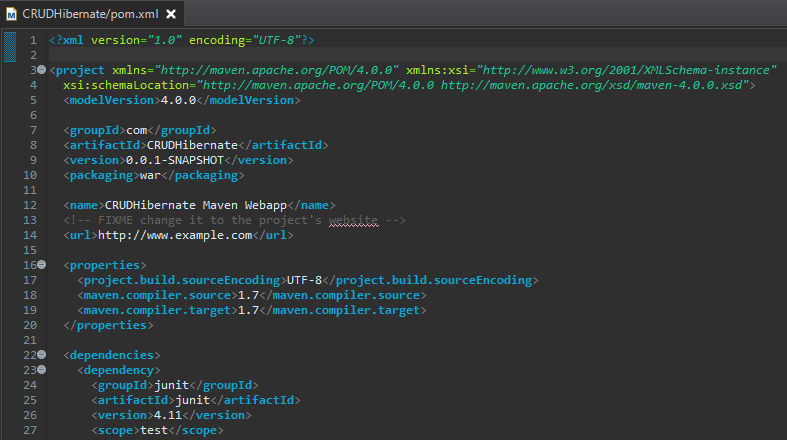
<**servlet-name**>ServiceWS</**servlet-name**>

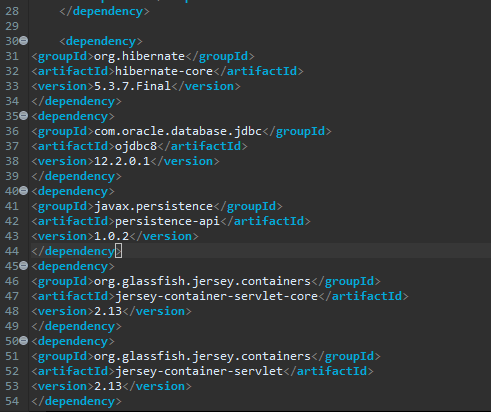
<**url-pattern**>/CRUDHibernate/\*</**url-pattern**>

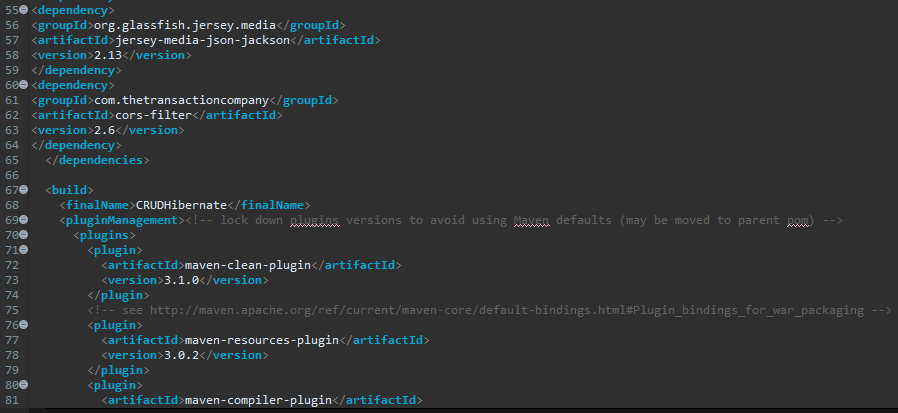
</**servlet-mapping**>

</**web-app**>

pom.xml









<?**xml** version=*"1.0"* encoding=*"UTF-8"*?>

<**project** xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<**modelVersion**>4.0.0</**modelVersion**>

<**groupId**>com</**groupId**>

<**artifactId**>CRUDHibernate</**artifactId**>

<**version**>0.0.1-SNAPSHOT</**version**>

<**packaging**>war</**packaging**>

<**name**>CRUDHibernate Maven Webapp</**name**>

<!-- FIXME change it to the project's website -->

<**url**>http://www.example.com</**url**>

<**properties**>

<**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>

<**maven.compiler.source**>1.7</**maven.compiler.source**>

<**maven.compiler.target**>1.7</**maven.compiler.target**>

</**properties**>

<**dependencies**>

<**dependency**>

<**groupId**>junit</**groupId**>

<**artifactId**>junit</**artifactId**>

<**version**>4.11</**version**>

<**scope**>test</**scope**>

</**dependency**>

<**dependency**>

<**groupId**>org.hibernate</**groupId**>

<**artifactId**>hibernate-core</**artifactId**>

<**version**>5.3.7.Final</**version**>

</**dependency**>

<**dependency**>

<**groupId**>com.oracle.database.jdbc</**groupId**>

<**artifactId**>ojdbc8</**artifactId**>

<**version**>12.2.0.1</**version**>

</**dependency**>

<**dependency**>

<**groupId**>javax.persistence</**groupId**>

<**artifactId**>persistence-api</**artifactId**>

<**version**>1.0.2</**version**>

</**dependency**>

<**dependency**>

<**groupId**>org.glassfish.jersey.containers</**groupId**>

<**artifactId**>jersey-container-servlet-core</**artifactId**>

<**version**>2.13</**version**>

</**dependency**>

<**dependency**>

<**groupId**>org.glassfish.jersey.containers</**groupId**>

<**artifactId**>jersey-container-servlet</**artifactId**>

<**version**>2.13</**version**>

</**dependency**>

<**dependency**>

<**groupId**>org.glassfish.jersey.media</**groupId**>

<**artifactId**>jersey-media-json-jackson</**artifactId**>

<**version**>2.13</**version**>

</**dependency**>

<**dependency**>

<**groupId**>com.thetransactioncompany</**groupId**>

<**artifactId**>cors-filter</**artifactId**>

<**version**>2.6</**version**>

</**dependency**>

</**dependencies**>

<**build**>

<**finalName**>CRUDHibernate</**finalName**>

<**pluginManagement**><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->

<**plugins**>

<**plugin**>

<**artifactId**>maven-clean-plugin</**artifactId**>

<**version**>3.1.0</**version**>

</**plugin**>

<!-- see http://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_war\_packaging -->

<**plugin**>

<**artifactId**>maven-resources-plugin</**artifactId**>

<**version**>3.0.2</**version**>

</**plugin**>

<**plugin**>

<**artifactId**>maven-compiler-plugin</**artifactId**>

<**version**>3.8.0</**version**>

</**plugin**>

<**plugin**>

<**artifactId**>maven-surefire-plugin</**artifactId**>

<**version**>2.22.1</**version**>

</**plugin**>

<**plugin**>

<**artifactId**>maven-war-plugin</**artifactId**>

<**version**>3.2.2</**version**>

</**plugin**>

<**plugin**>

<**artifactId**>maven-install-plugin</**artifactId**>

<**version**>2.5.2</**version**>

</**plugin**>

<**plugin**>

<**artifactId**>maven-deploy-plugin</**artifactId**>

<**version**>2.8.2</**version**>

</**plugin**>

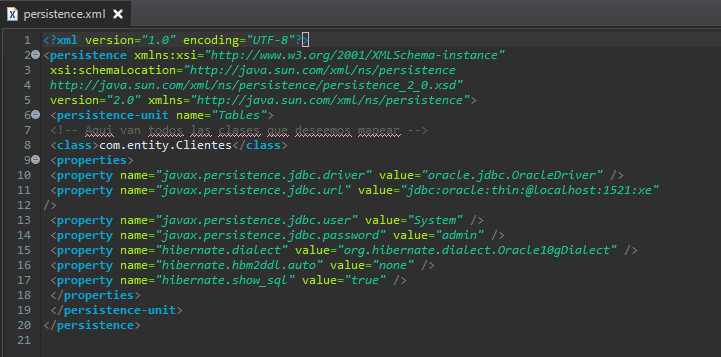
</**plugins**>

</**pluginManagement**>

</**build**>

</**project**>

persistence.xml



<?**xml** version=*"1.0"* encoding=*"UTF-8"*?>

<**persistence** xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/persistence*

*http://java.sun.com/xml/ns/persistence/persistence\_2\_0.xsd"*

version=*"2.0"* xmlns=*"http://java.sun.com/xml/ns/persistence"*>

<**persistence-unit** name=*"Tables"*>

<!-- Aqui van todos las clases que deseemos mapear -->

<**class**>com.entity.Clientes</**class**>

<**properties**>

<**property** name=*"javax.persistence.jdbc.driver"* value=*"oracle.jdbc.OracleDriver"* />

<**property** name=*"javax.persistence.jdbc.url"* value=*"jdbc:oracle:thin:@localhost:1521:xe"*

/>

<**property** name=*"javax.persistence.jdbc.user"* value=*"System"* />

<**property** name=*"javax.persistence.jdbc.password"* value=*"admin"* />

<**property** name=*"hibernate.dialect"* value=*"org.hibernate.dialect.Oracle10gDialect"* />

<**property** name=*"hibernate.hbm2ddl.auto"* value=*"none"* />

<**property** name=*"hibernate.show\_sql"* value=*"true"* />

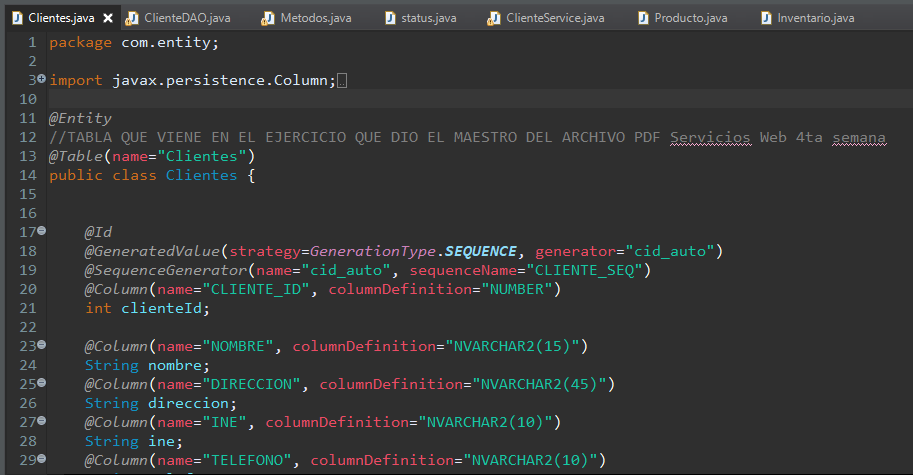
</**properties**>

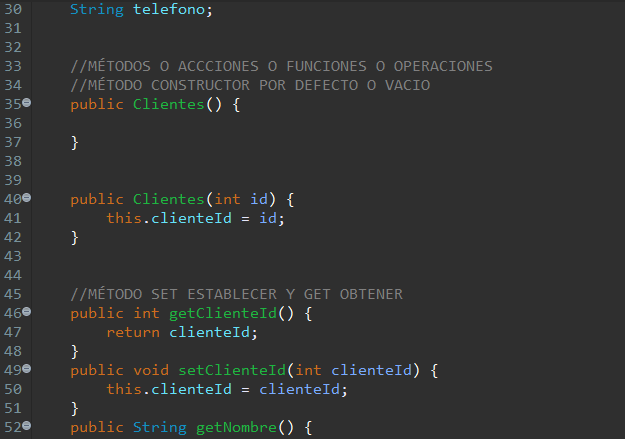
</**persistence-unit**>

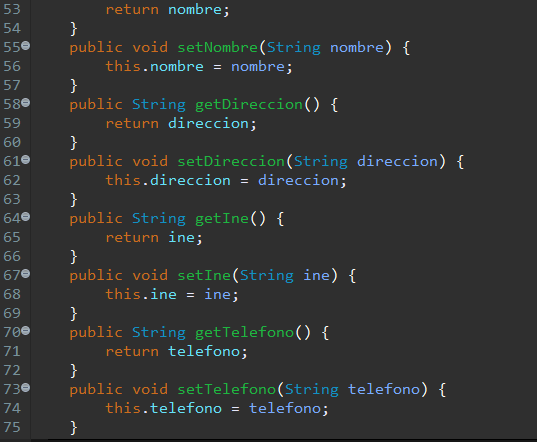
</**persistence**>

Clase: Clientes

Package: com.entity









package com.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.SequenceGenerator;

import javax.persistence.Table;

@Entity

//TABLA QUE VIENE EN EL EJERCICIO QUE DIO EL MAESTRO DEL ARCHIVO PDF Servicios Web 4ta semana

@Table(name="Clientes")

public class Clientes {

@Id

@GeneratedValue(strategy=GenerationType.SEQUENCE, generator="cid\_auto")

@SequenceGenerator(name="cid\_auto", sequenceName="CLIENTE\_SEQ")

@Column(name="CLIENTE\_ID", columnDefinition="NUMBER")

int clienteId;

@Column(name="NOMBRE", columnDefinition="NVARCHAR2(15)")

String nombre;

@Column(name="DIRECCION", columnDefinition="NVARCHAR2(45)")

String direccion;

@Column(name="INE", columnDefinition="NVARCHAR2(10)")

String ine;

@Column(name="TELEFONO", columnDefinition="NVARCHAR2(10)")

String telefono;

//MÉTODOS O ACCCIONES O FUNCIONES O OPERACIONES

//MÉTODO CONSTRUCTOR POR DEFECTO O VACIO

public Clientes() {

}

public Clientes(int id) {

this.clienteId = id;

}

//MÉTODO SET ESTABLECER Y GET OBTENER

public int getClienteId() {

return clienteId;

}

public void setClienteId(int clienteId) {

this.clienteId = clienteId;

}

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public String getDireccion() {

return direccion;

}

public void setDireccion(String direccion) {

this.direccion = direccion;

}

public String getIne() {

return ine;

}

public void setIne(String ine) {

this.ine = ine;

}

public String getTelefono() {

return telefono;

}

public void setTelefono(String telefono) {

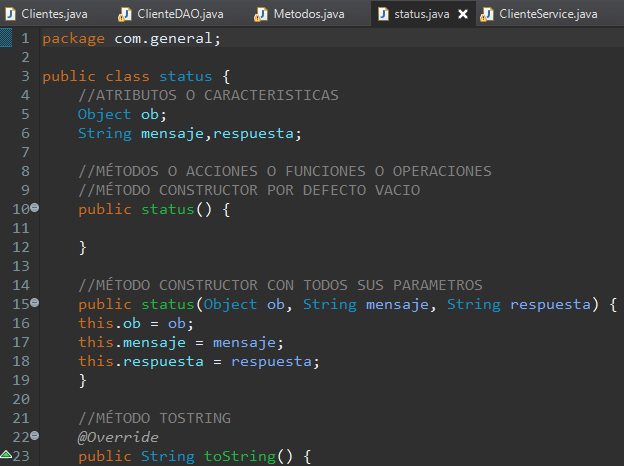
this.telefono = telefono;

}

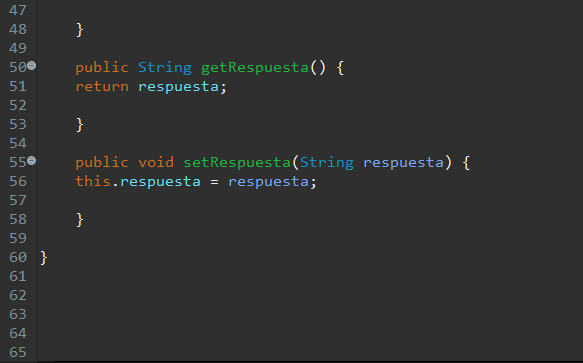
}

Clase: status

Package: com.general







package com.general;

public class status {

//ATRIBUTOS O CARACTERISTICAS

Object ob;

String mensaje,respuesta;

//MÉTODOS O ACCIONES O FUNCIONES O OPERACIONES

//MÉTODO CONSTRUCTOR POR DEFECTO VACIO

public status() {

}

//MÉTODO CONSTRUCTOR CON TODOS SUS PARAMETROS

public status(Object ob, String mensaje, String respuesta) {

this.ob = ob;

this.mensaje = mensaje;

this.respuesta = respuesta;

}

//MÉTODO TOSTRING

*@Override*

public String toString() {

return "Status [ob=" + ob + ", mensaje=" + mensaje + ", respuesta=" + respuesta +

"]\n";

}

//MÉTODO SET ESTABLECER Y GET OBTENER

public Object getOb() {

return ob;

}

public void setOb(Object ob) {

this.ob = ob;

}

public String getMensaje() {

return mensaje;

}

public void setMensaje(String mensaje) {

this.mensaje = mensaje;

}

public String getRespuesta() {

return respuesta;

}

public void setRespuesta(String respuesta) {

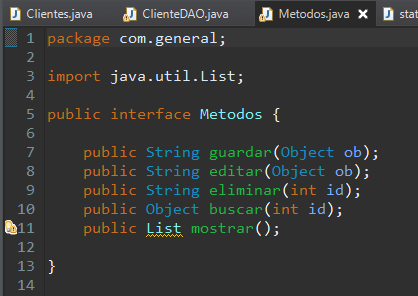
this.respuesta = respuesta;

}

}

Interfase: Metodos

Package: com.general



package com.general;

import java.util.List;

public interface Metodos {

public String guardar(Object ob);

public String editar(Object ob);

public String eliminar(int id);

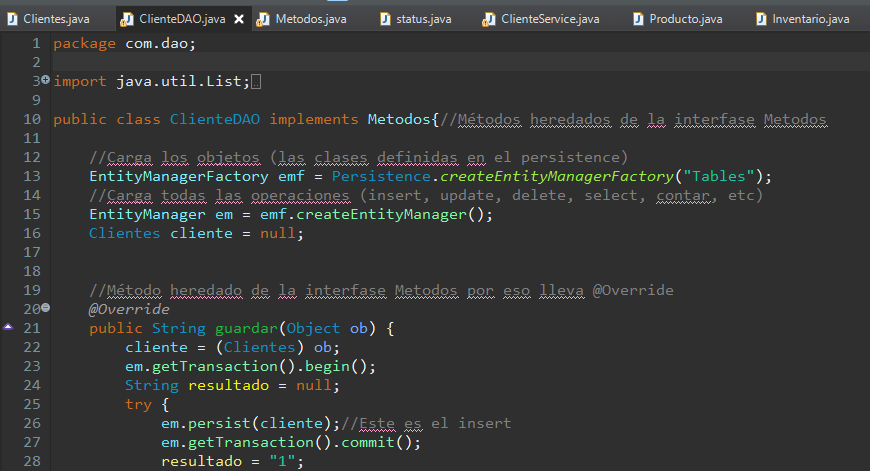
public Object buscar(int id);

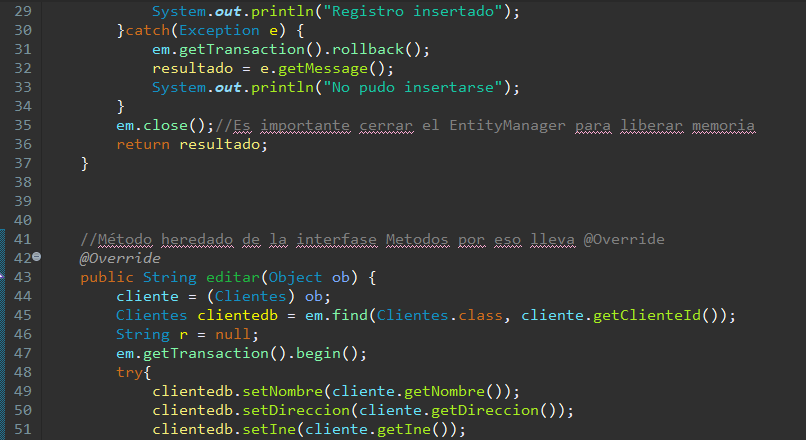
public List mostrar();

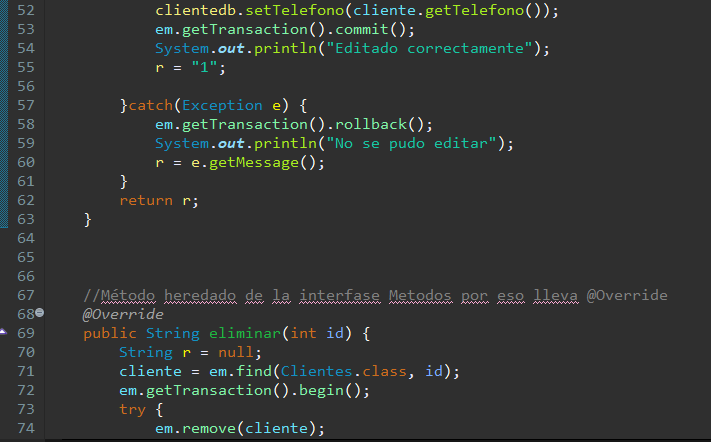
}

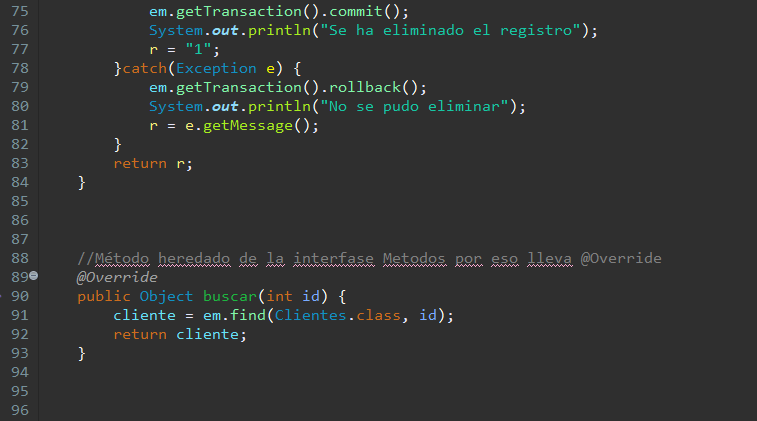
Clase: ClienteDAO

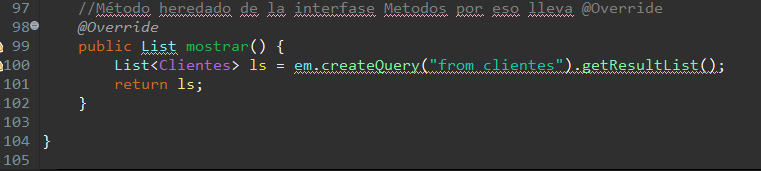
Package: com.dao











package com.dao;

import java.util.List;

import javax.persistence.EntityManager;

import javax.persistence.EntityManagerFactory;

import javax.persistence.Persistence;

import com.entity.Clientes;

import com.general.Metodos;

public class ClienteDAO implements Metodos{//Métodos heredados de la interfase Metodos

//Carga los objetos (las clases definidas en el persistence)

EntityManagerFactory emf = Persistence.createEntityManagerFactory("Tables");

//Carga todas las operaciones (insert, update, delete, select, contar, etc)

EntityManager em = emf.createEntityManager();

Clientes cliente = null;

//Método heredado de la interfase Metodos por eso lleva @Override

@Override

public String guardar(Object ob) {

cliente = (Clientes) ob;

em.getTransaction().begin();

String resultado = null;

try {

em.persist(cliente);//Este es el insert

em.getTransaction().commit();

resultado = "1";

System.out.println("Registro insertado");

}catch(Exception e) {

em.getTransaction().rollback();

resultado = e.getMessage();

System.out.println("No pudo insertarse");

}

em.close();//Es importante cerrar el EntityManager para liberar memoria

return resultado;

}

//Método heredado de la interfase Metodos por eso lleva @Override

@Override

public String editar(Object ob) {

cliente = (Clientes) ob;

Clientes clientedb = em.find(Clientes.class, cliente.getClienteId());

String r = null;

em.getTransaction().begin();

try{

clientedb.setNombre(cliente.getNombre());

clientedb.setDireccion(cliente.getDireccion());

clientedb.setIne(cliente.getIne());

clientedb.setTelefono(cliente.getTelefono());

em.getTransaction().commit();

System.out.println("Editado correctamente");

r = "1";

}catch(Exception e) {

em.getTransaction().rollback();

System.out.println("No se pudo editar");

r = e.getMessage();

}

return r;

}

//Método heredado de la interfase Metodos por eso lleva @Override

@Override

public String eliminar(int id) {

String r = null;

cliente = em.find(Clientes.class, id);

em.getTransaction().begin();

try {

em.remove(cliente);

em.getTransaction().commit();

System.out.println("Se ha eliminado el registro");

r = "1";

}catch(Exception e) {

em.getTransaction().rollback();

System.out.println("No se pudo eliminar");

r = e.getMessage();

}

return r;

}

//Método heredado de la interfase Metodos por eso lleva @Override

@Override

public Object buscar(int id) {

cliente = em.find(Clientes.class, id);

return cliente;

}

//Método heredado de la interfase Metodos por eso lleva @Override

@Override

public List mostrar() {

List<Clientes> ls = em.createQuery("from clientes").getResultList();

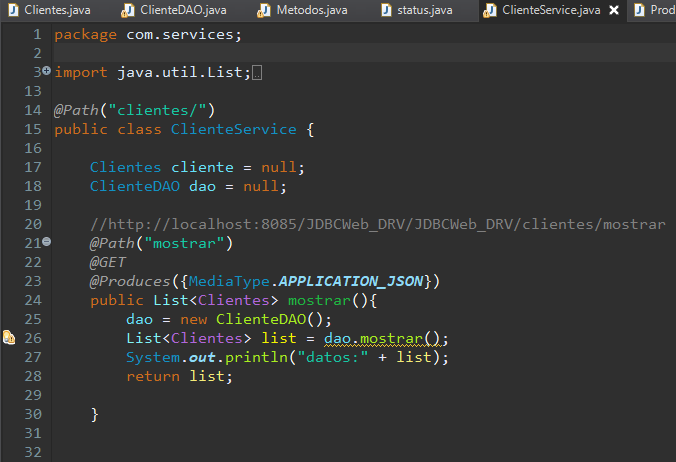
return ls;

}

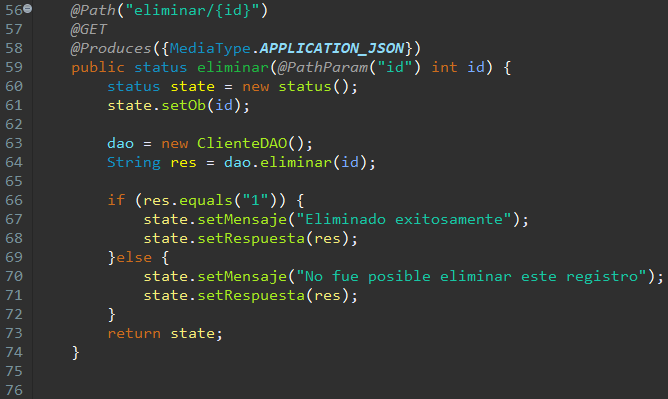
}

Clase: ClienteService

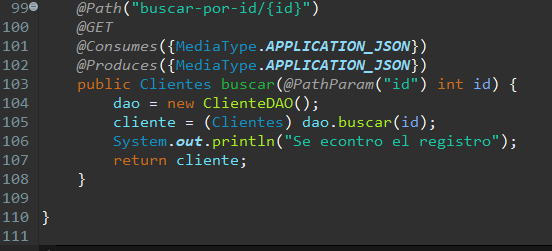
Package: com.services











package com.services;

import java.util.List;

import javax.ws.rs.Consumes;

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.PathParam;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

import com.dao.ClienteDAO;

import com.entity.Clientes;

import com.general.status;

@Path("clientes/")

public class ClienteService {

Clientes cliente = null;

ClienteDAO dao = null;

//http://localhost:8085/JDBCWeb\_DRV/JDBCWeb\_DRV/clientes/mostrar

@Path("mostrar")

@GET

@Produces({MediaType.APPLICATION\_JSON})

public List<Clientes> mostrar(){

dao = new ClienteDAO();

List<Clientes> list = dao.mostrar();

System.out.println("datos:" + list);

return list;

}

@Path("guardar")

@GET

@Consumes({MediaType.APPLICATION\_JSON})

@Produces({MediaType.APPLICATION\_JSON})

public status guardar(Clientes c) {

System.out.println("cliente a guardar: " + c);

status state = new status();

state.setOb(c);

dao = new ClienteDAO();

String res = dao.guardar(c);

if (res.equals("1")) {

state.setMensaje("Guardado exitosamente");

state.setRespuesta(res);

}else {

state.setMensaje("No fue posible guardar");

state.setRespuesta(res);

}

return state;

}

@Path("eliminar/{id}")

@GET

@Produces({MediaType.APPLICATION\_JSON})

public status eliminar(@PathParam("id") int id) {

status state = new status();

state.setOb(id);

dao = new ClienteDAO();

String res = dao.eliminar(id);

if (res.equals("1")) {

state.setMensaje("Eliminado exitosamente");

state.setRespuesta(res);

}else {

state.setMensaje("No fue posible eliminar este registro");

state.setRespuesta(res);

}

return state;

}

@Path("editar")

@GET

@Consumes({MediaType.APPLICATION\_JSON})

@Produces({MediaType.APPLICATION\_JSON})

public status editar(Clientes c) {

status state = new status();

state.setOb(c);

dao = new ClienteDAO();

String res = dao.editar(c);

if (res.equals("1")) {

state.setMensaje("Actualizado correctamente");

state.setRespuesta(res);

}else {

state.setMensaje("No fue posible actualizar este registro");

state.setRespuesta(res);

}

return state;

}

@Path("buscar-por-id/{id}")

@GET

@Consumes({MediaType.APPLICATION\_JSON})

@Produces({MediaType.APPLICATION\_JSON})

public Clientes buscar(@PathParam("id") int id) {

dao = new ClienteDAO();

cliente = (Clientes) dao.buscar(id);

System.out.println("Se econtro el registro");

return cliente;

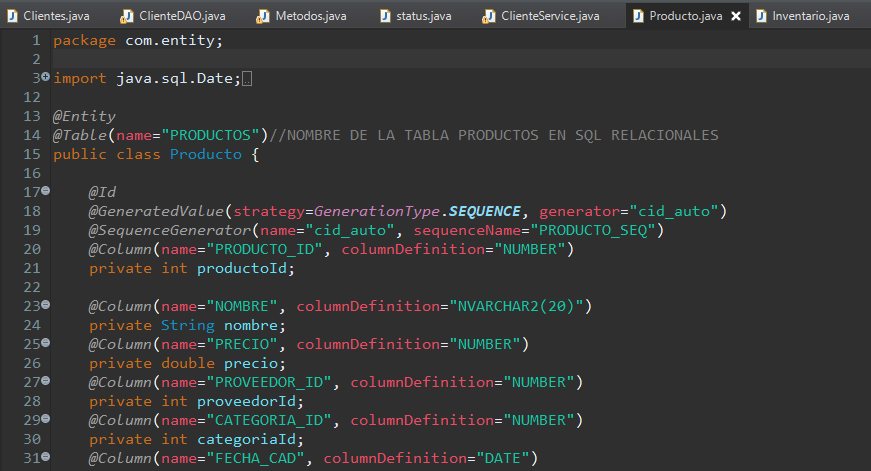
}

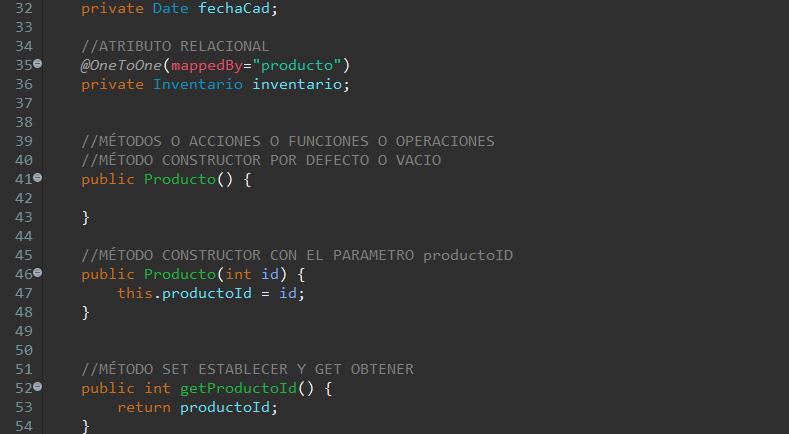
}

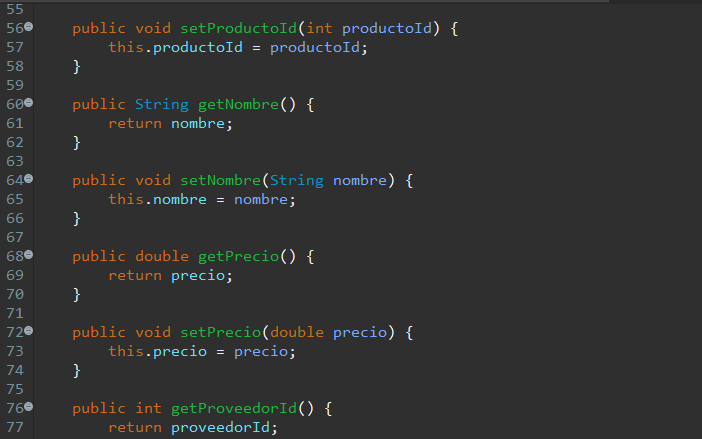
Clase: Producto

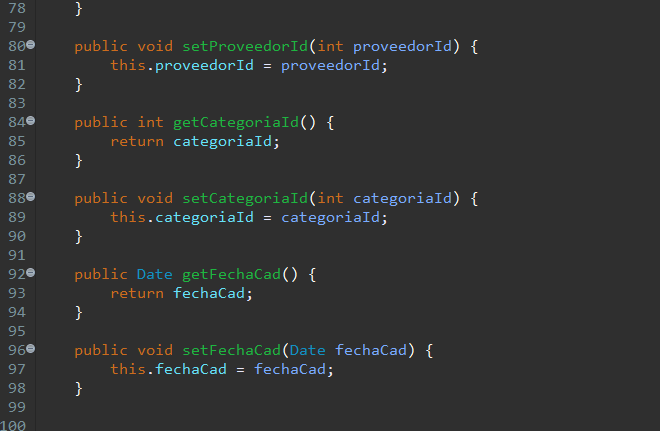
Package: com.entity

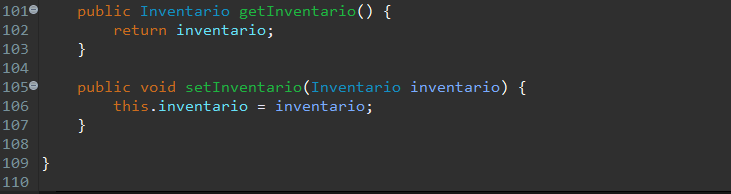
Falto colocar el atributo presentacion de la tabla PRODUCTOS en SQL RELACIONALES











package com.entity;

import java.sql.Date;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToOne;

import javax.persistence.SequenceGenerator;

import javax.persistence.Table;

@Entity

@Table(name="PRODUCTOS")//NOMBRE DE LA TABLA PRODUCTOS EN SQL RELACIONALES

public class Producto {

@Id

@GeneratedValue(strategy=GenerationType.SEQUENCE, generator="cid\_auto")

@SequenceGenerator(name="cid\_auto", sequenceName="PRODUCTO\_SEQ")

@Column(name="PRODUCTO\_ID", columnDefinition="NUMBER")

private int productoId;

@Column(name="NOMBRE", columnDefinition="NVARCHAR2(20)")

private String nombre;

@Column(name="PRECIO", columnDefinition="NUMBER")

private double precio;

@Column(name="PROVEEDOR\_ID", columnDefinition="NUMBER")

private int proveedorId;

@Column(name="CATEGORIA\_ID", columnDefinition="NUMBER")

private int categoriaId;

@Column(name="FECHA\_CAD", columnDefinition="DATE")

private Date fechaCad;

//ATRIBUTO RELACIONAL

@OneToOne(mappedBy="producto")

private Inventario inventario;

//MÉTODOS O ACCIONES O FUNCIONES O OPERACIONES

//MÉTODO CONSTRUCTOR POR DEFECTO O VACIO

public Producto() {

}

//MÉTODO CONSTRUCTOR CON EL PARAMETRO productoID

public Producto(int id) {

this.productoId = id;

}

//MÉTODO SET ESTABLECER Y GET OBTENER

public int getProductoId() {

return productoId;

}

public void setProductoId(int productoId) {

this.productoId = productoId;

}

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public double getPrecio() {

return precio;

}

public void setPrecio(double precio) {

this.precio = precio;

}

public int getProveedorId() {

return proveedorId;

}

public void setProveedorId(int proveedorId) {

this.proveedorId = proveedorId;

}

public int getCategoriaId() {

return categoriaId;

}

public void setCategoriaId(int categoriaId) {

this.categoriaId = categoriaId;

}

public Date getFechaCad() {

return fechaCad;

}

public void setFechaCad(Date fechaCad) {

this.fechaCad = fechaCad;

}

public Inventario getInventario() {

return inventario;

}

public void setInventario(Inventario inventario) {

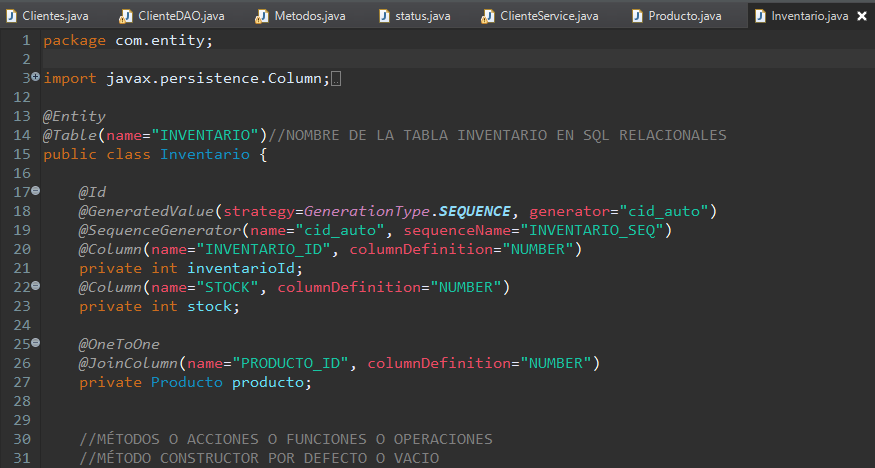
this.inventario = inventario;

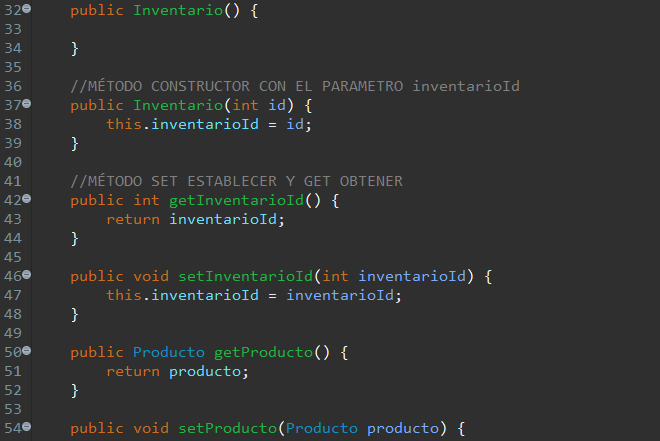
}

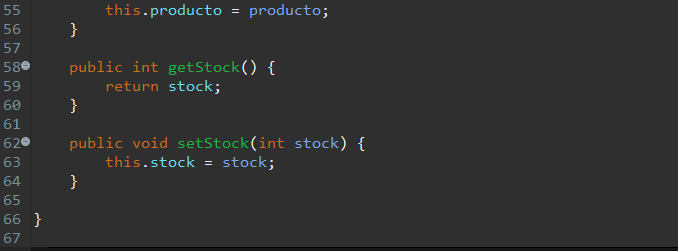
}

Clase: Inventario

Package: com.entity







package com.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToOne;

import javax.persistence.SequenceGenerator;

import javax.persistence.Table;

@Entity

@Table(name="INVENTARIO")//NOMBRE DE LA TABLA INVENTARIO EN SQL RELACIONALES

public class Inventario {

@Id

@GeneratedValue(strategy=GenerationType.SEQUENCE, generator="cid\_auto")

@SequenceGenerator(name="cid\_auto", sequenceName="INVENTARIO\_SEQ")

@Column(name="INVENTARIO\_ID", columnDefinition="NUMBER")

private int inventarioId;

@Column(name="STOCK", columnDefinition="NUMBER")

private int stock;

@OneToOne

@JoinColumn(name="PRODUCTO\_ID", columnDefinition="NUMBER")

private Producto producto;

//MÉTODOS O ACCIONES O FUNCIONES O OPERACIONES

//MÉTODO CONSTRUCTOR POR DEFECTO O VACIO

public Inventario() {

}

//MÉTODO CONSTRUCTOR CON EL PARAMETRO inventarioId

public Inventario(int id) {

this.inventarioId = id;

}

//MÉTODO SET ESTABLECER Y GET OBTENER

public int getInventarioId() {

return inventarioId;

}

public void setInventarioId(int inventarioId) {

this.inventarioId = inventarioId;

}

public Producto getProducto() {

return producto;

}

public void setProducto(Producto producto) {

this.producto = producto;

}

public int getStock() {

return stock;

}

public void setStock(int stock) {

this.stock = stock;

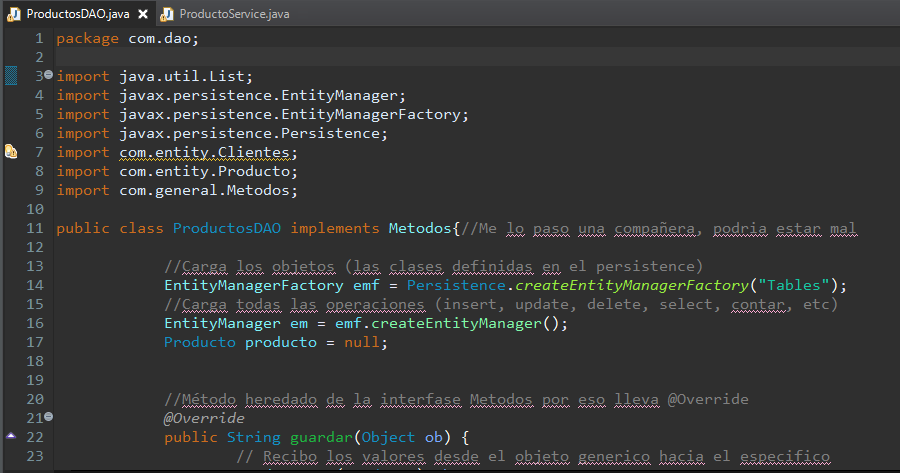
}

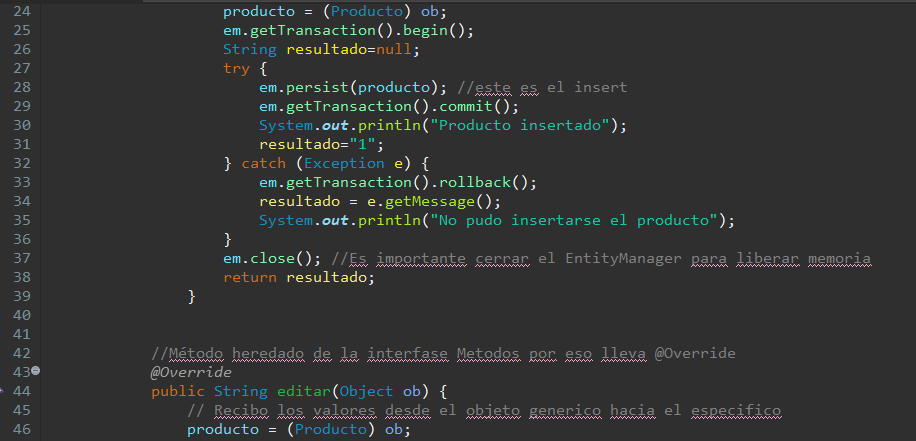
}

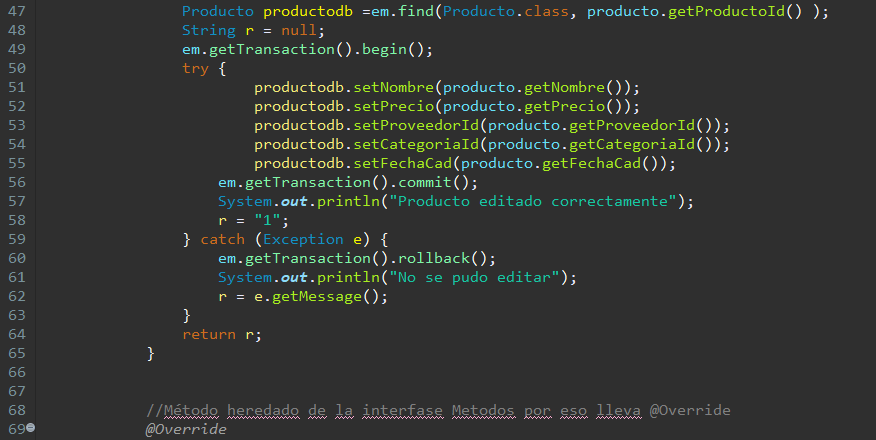
Me lo paso una compañera

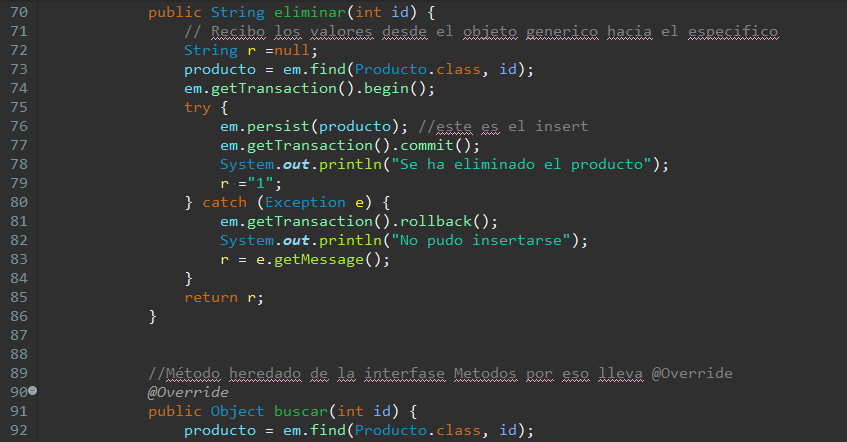
Clase: ProductosDAO

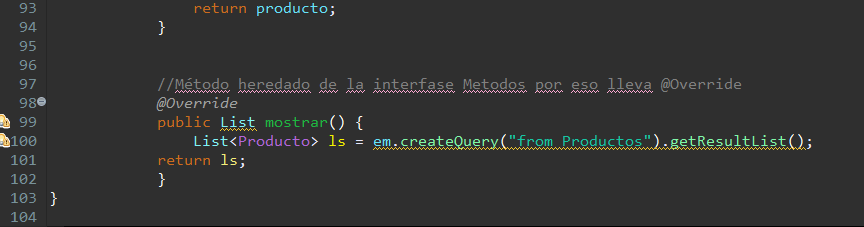
Package: com.dao











package com.dao;

import java.util.List;

import javax.persistence.EntityManager;

import javax.persistence.EntityManagerFactory;

import javax.persistence.Persistence;

import com.entity.Clientes;

import com.entity.Producto;

import com.general.Metodos;

public class ProductosDAO implements Metodos{//Me lo paso una compañera, podria estar mal

//Carga los objetos (las clases definidas en el persistence)

EntityManagerFactory emf = Persistence.*createEntityManagerFactory*("Tables");

//Carga todas las operaciones (insert, update, delete, select, contar, etc)

EntityManager em = emf.createEntityManager();

Producto producto = null;

//Método heredado de la interfase Metodos por eso lleva @Override

*@Override*

public String guardar(Object ob) {

// Recibo los valores desde el objeto generico hacia el especifico

producto = (Producto) ob;

em.getTransaction().begin();

String resultado=null;

try {

em.persist(producto); //este es el insert

em.getTransaction().commit();

System.***out***.println("Producto insertado");

resultado="1";

} catch (Exception e) {

em.getTransaction().rollback();

resultado = e.getMessage();

System.***out***.println("No pudo insertarse el producto");

}

em.close(); //Es importante cerrar el EntityManager para liberar memoria

return resultado;

}

//Método heredado de la interfase Metodos por eso lleva @Override

*@Override*

public String editar(Object ob) {

// Recibo los valores desde el objeto generico hacia el especifico

producto = (Producto) ob;

Producto productodb =em.find(Producto.class, producto.getProductoId() );

String r = null;

em.getTransaction().begin();

try {

productodb.setNombre(producto.getNombre());

productodb.setPrecio(producto.getPrecio());

productodb.setProveedorId(producto.getProveedorId());

productodb.setCategoriaId(producto.getCategoriaId());

productodb.setFechaCad(producto.getFechaCad());

em.getTransaction().commit();

System.***out***.println("Producto editado correctamente");

r = "1";

} catch (Exception e) {

em.getTransaction().rollback();

System.***out***.println("No se pudo editar");

r = e.getMessage();

}

return r;

}

//Método heredado de la interfase Metodos por eso lleva @Override

*@Override*

public String eliminar(int id) {

// Recibo los valores desde el objeto generico hacia el especifico

String r =null;

producto = em.find(Producto.class, id);

em.getTransaction().begin();

try {

em.persist(producto); //este es el insert

em.getTransaction().commit();

System.***out***.println("Se ha eliminado el producto");

r ="1";

} catch (Exception e) {

em.getTransaction().rollback();

System.***out***.println("No pudo insertarse");

r = e.getMessage();

}

return r;

}

//Método heredado de la interfase Metodos por eso lleva @Override

*@Override*

public Object buscar(int id) {

producto = em.find(Producto.class, id);

return producto;

}

//Método heredado de la interfase Metodos por eso lleva @Override

*@Override*

public List mostrar() {

List<Producto> ls = em.createQuery("from Productos").getResultList();

return ls;

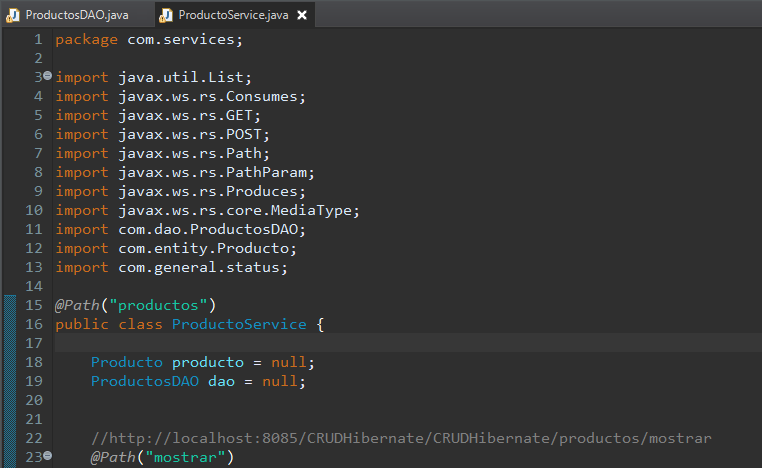
}

}

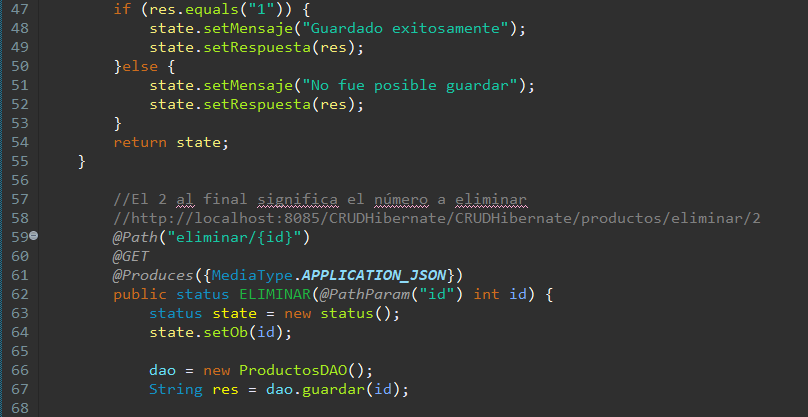
Me lo paso una compañera

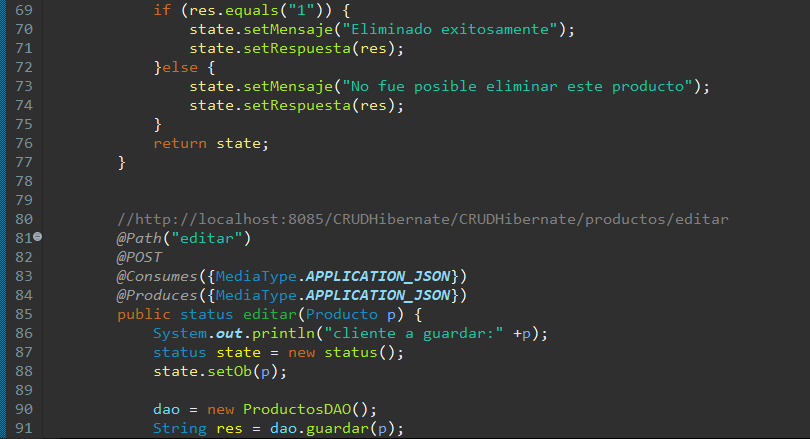
Clase: ProductoService

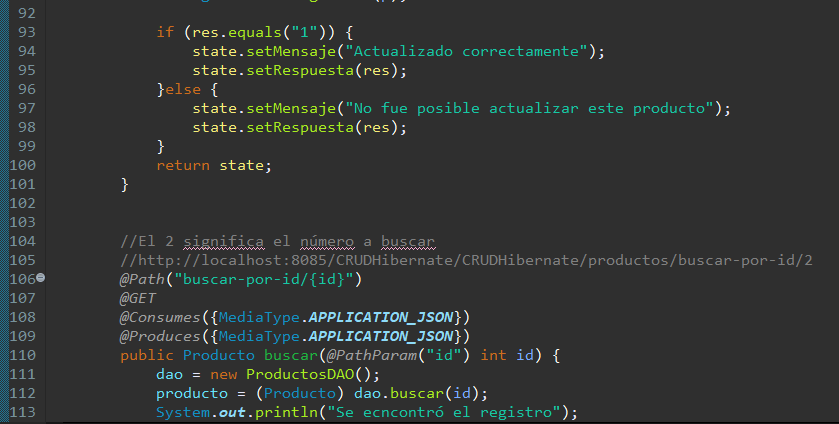
Package: com.services













package com.services;

import java.util.List;

import javax.ws.rs.Consumes;

import javax.ws.rs.GET;

import javax.ws.rs.POST;

import javax.ws.rs.Path;

import javax.ws.rs.PathParam;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

import com.dao.ProductosDAO;

import com.entity.Producto;

import com.general.status;

*@Path*("productos")

public class ProductoService {

Producto producto = null;

ProductosDAO dao = null;

//http://localhost:8085/CRUDHibernate/CRUDHibernate/productos/mostrar

*@Path*("mostrar")

*@GET*

*@Produces*({MediaType.***APPLICATION\_JSON***})

public List<Producto> mostrar() {

dao = new ProductosDAO();

List <Producto> list = dao.mostrar();

System.***out***.println("datos:" + list);

return list;

}

//http://localhost:8085/CRUDHibernate/CRUDHibernate/productos/guardar

*@Path*("guardar")

*@POST*

*@Consumes*({MediaType.***APPLICATION\_JSON***})

*@Produces*({MediaType.***APPLICATION\_JSON***})

public status guardar(Producto p) {

System.***out***.println("Producto a guardar:" +p);

status state = new status();

state.setOb(p);

dao = new ProductosDAO();

String res = dao.guardar(p);

if (res.equals("1")) {

state.setMensaje("Guardado exitosamente");

state.setRespuesta(res);

}else {

state.setMensaje("No fue posible guardar");

state.setRespuesta(res);

}

return state;

}

//El 2 al final significa el número a eliminar

//http://localhost:8085/CRUDHibernate/CRUDHibernate/productos/eliminar/2

*@Path*("eliminar/{id}")

*@GET*

*@Produces*({MediaType.***APPLICATION\_JSON***})

public status ELIMINAR(*@PathParam*("id") int id) {

status state = new status();

state.setOb(id);

dao = new ProductosDAO();

String res = dao.guardar(id);

if (res.equals("1")) {

state.setMensaje("Eliminado exitosamente");

state.setRespuesta(res);

}else {

state.setMensaje("No fue posible eliminar este producto");

state.setRespuesta(res);

}

return state;

}

//http://localhost:8085/CRUDHibernate/CRUDHibernate/productos/editar

*@Path*("editar")

*@POST*

*@Consumes*({MediaType.***APPLICATION\_JSON***})

*@Produces*({MediaType.***APPLICATION\_JSON***})

public status editar(Producto p) {

System.***out***.println("cliente a guardar:" +p);

status state = new status();

state.setOb(p);

dao = new ProductosDAO();

String res = dao.guardar(p);

if (res.equals("1")) {

state.setMensaje("Actualizado correctamente");

state.setRespuesta(res);

}else {

state.setMensaje("No fue posible actualizar este producto");

state.setRespuesta(res);

}

return state;

}

//El 2 significa el número a buscar

//http://localhost:8085/CRUDHibernate/CRUDHibernate/productos/buscar-por-id/2

*@Path*("buscar-por-id/{id}")

*@GET*

*@Consumes*({MediaType.***APPLICATION\_JSON***})

*@Produces*({MediaType.***APPLICATION\_JSON***})

public Producto buscar(*@PathParam*("id") int id) {

dao = new ProductosDAO();

producto = (Producto) dao.buscar(id);

System.***out***.println("Se ecncontró el registro");

return producto;

}

}