DAO.ClienteDao

package dao;

import java.util.\*;

import org.hibernate.Hibernate;

import org.hibernate.HibernateException;

import org.hibernate.Session;

import org.hibernate.Transaction;

import datos.Cliente;

public class ClienteDao {

private static Session session;

private Transaction tx;

private void iniciaOperacion() throws HibernateException {

session = HibernateUtil.getSessionFactory().openSession();

tx = session.beginTransaction();

}

private void manejaExcepcion(HibernateException he) throws HibernateException {

tx.rollback();

throw new HibernateException("ERROR en la capa de acceso a datos", he);

}

public int alta(Cliente objeto) {

int id = 0;

try {

iniciaOperacion();

id = Integer.parseInt(session.save(objeto).toString());

tx.commit();

} catch (HibernateException he) {

manejaExcepcion(he);

throw he;

} finally {

session.close();

}

return id;

}

public void modificacion(Cliente objeto) throws HibernateException {

try {

iniciaOperacion();

session.update(objeto);

tx.commit();

} catch (HibernateException he) {

manejaExcepcion(he);

throw he;

} finally {

session.close();

}

}

public void baja(Cliente objeto) throws HibernateException {

try {

iniciaOperacion();

session.delete(objeto);

tx.commit();

} catch (HibernateException he) {

manejaExcepcion(he);

throw he;

} finally {

session.close();

}

}

public Cliente traer(int idCliente) throws HibernateException {

Cliente objeto = null;

try {

iniciaOperacion();

objeto = (Cliente) session.get(Cliente.class, idCliente);

} finally {

session.close();

}

return objeto;

}

public Cliente traer(long dni) throws HibernateException {

Cliente objeto = null;

try {

iniciaOperacion();

objeto = (Cliente) session.createQuery("from Cliente c where c.dni="+dni).uniqueResult();

} finally {

session.close();

}

return objeto;

}

@SuppressWarnings("unchecked")

public List<Cliente> traer() throws HibernateException {

List<Cliente> lista=null;

try {

iniciaOperacion();

lista=session.createQuery("from Cliente c order by c.apellido asc,c.nombre asc").list();

} finally {

session.close();

}

return lista;

}

//13

public Cliente traer(String email) throws HibernateException {

Cliente objeto = null;

try {

iniciaOperacion();

objeto = (Cliente) session.createQuery("from Cliente c where c.email="+email).uniqueResult();

} finally {

session.close();

}

return objeto;

}

}

DAO.EmpleadoDao

package dao;

import java.util.List;

import org.hibernate.HibernateException;

import org.hibernate.Session;

import org.hibernate.Transaction;

import datos.Empleado;

public class EmpleadoDao {

private static Session session;

private Transaction tx;

private void iniciaOperacion() throws HibernateException {

session = HibernateUtil.getSessionFactory().openSession();

tx = session.beginTransaction();

}

private void manejaExcepcion(HibernateException he) throws HibernateException {

tx.rollback();

throw new HibernateException("ERROR en la capa de acceso a datos", he);

}

public int alta(Empleado objeto) {

int id = 0;

try {

iniciaOperacion();

id = Integer.parseInt(session.save(objeto).toString());

tx.commit();

} catch (HibernateException he) {

manejaExcepcion(he);

throw he;

} finally {

session.close();

}

return id;

}

public void modificacion(Empleado objeto) throws HibernateException {

try {

iniciaOperacion();

session.update(objeto);

tx.commit();

} catch (HibernateException he) {

manejaExcepcion(he);

throw he;

} finally {

session.close();

}

}

public void baja(Empleado objeto) throws HibernateException {

try {

iniciaOperacion();

session.delete(objeto);

tx.commit();

} catch (HibernateException he) {

manejaExcepcion(he);

throw he;

} finally {

session.close();

}

}

public Empleado traer(int idCliente) throws HibernateException {

Empleado objeto = null;

try {

iniciaOperacion();

objeto = (Empleado) session.get(Empleado.class, idCliente);

} finally {

session.close();

}

return objeto;

}

public Empleado traer(long dni) throws HibernateException {

Empleado objeto = null;

try {

iniciaOperacion();

objeto = (Empleado) session.createQuery("from Cliente c where c.dni="+dni).uniqueResult();

} finally {

session.close();

}

return objeto;

}

@SuppressWarnings("unchecked")

public List<Empleado> traer() throws HibernateException {

List<Empleado> lista=null;

try {

iniciaOperacion();

lista=session.createQuery("from Cliente c order by c.apellido asc,c.nombre asc").list();

} finally {

session.close();

}

return lista;

}

}

NEGOCIO.ClienteABM

package negocio;

import java.time.LocalDate;

import java.util.\*;

import datos.Cliente;

import dao.ClienteDao;

public class ClienteABM {

ClienteDao dao = new ClienteDao();

public Cliente traer(int idCliente) {

Cliente c = dao.traer(idCliente);

return c;

}

public Cliente traer(long dni) {

Cliente c = dao.traer(dni);

return c;

}

public Cliente traer(String email) {

Cliente c = dao.traer(email);

return c;

}

public int alta(long dni, String apellido, String nombre, LocalDate fechaDeNacimiento, String email)

throws Exception {

if ((Cliente) dao.traer(dni) != null || (Cliente) dao.traer(email) != null)

throw new Exception("Ya existe un cliente con ese dni");

Cliente c = new Cliente(dni, apellido, nombre, fechaDeNacimiento, email);

return dao.alta(c);

}

public void modificar(Cliente c) throws Exception {

if ((Cliente) dao.traer(c.getDni()) != null || (Cliente) dao.traer(c.getIdCliente()) != null)

throw new Exception("Ya existe un cliente con ese dni o idCliente");

dao.modificacion(c);

}

public void eliminar(int idCliente) {

Cliente c = dao.traer(idCliente);

dao.baja(c);

}

public List<Cliente> traer() {

return dao.traer();

}

}

NEGOCIO.EmpleadoABM

package negocio;

import java.time.LocalDate;

import java.util.List;

import datos.Local;

import java.time.\*;

import dao.ClienteDao;

import dao.EmpleadoDao;

import datos.Cliente;

import datos.Empleado;

import java.util.\*;

public class EmpleadoABM {

EmpleadoDao dao = new EmpleadoDao();

public Empleado traer(int idCliente) {

Empleado e = dao.traer(idCliente);

return e;

}

public Empleado traer(long dni) {

Empleado e = dao.traer(dni);

return e;

}

public int alta(long dni, String apellido, String nombre, LocalDate fechaDeNacimiento,float sueldoBasico,float comision,LocalTime horarioTrabajoE,LocalTime horarioTrabajos,boolean gerente,Local local )

throws Exception {

if ((Empleado) dao.traer(dni) != null)

throw new Exception("Ya existe un empleado con ese dni");

Empleado e = new Empleado(dni, apellido, nombre, fechaDeNacimiento, sueldoBasico, comision, horarioTrabajoE, horarioTrabajos, gerente, local );

return dao.alta(e);

}

public void modificar(Empleado e) throws Exception {

if ((Empleado) dao.traer(e.getDni()) != null || (Empleado) dao.traer(e.getId()) != null)

throw new Exception("Ya existe un empleado con ese dni o idEmpleado");

dao.modificacion(e);

}

public void eliminar(long idCliente) {

Empleado e = dao.traer(idCliente);

dao.baja(e);

}

public List<Empleado> traer() {

return dao.traer();

}

}