

Carrera: Licenciatura en Sistemas

Materia: Orientación a Objetos II

Equipo docente:

Titular: Prof. María Alejandra Vranić

[alejandravranic@gmail.com](mailto:alejandravranic@gmail.com)

Ayudantes: Prof. Leandro Ríos

[leandro.rios.unla@gmail.com](mailto:leandro.rios.unla@gmail.com)

Prof. Gustavo Siciliano

[gussiciliano@gmail.com](mailto:gussiciliano@gmail.com)

Prof. Romina Mansilla

[romina.e.mansilla@gmail.com](mailto:romina.e.mansilla@gmail.com)



Año: 2018

IDE: Eclipse

Persistencia de datos: feriado.xml, MySQL

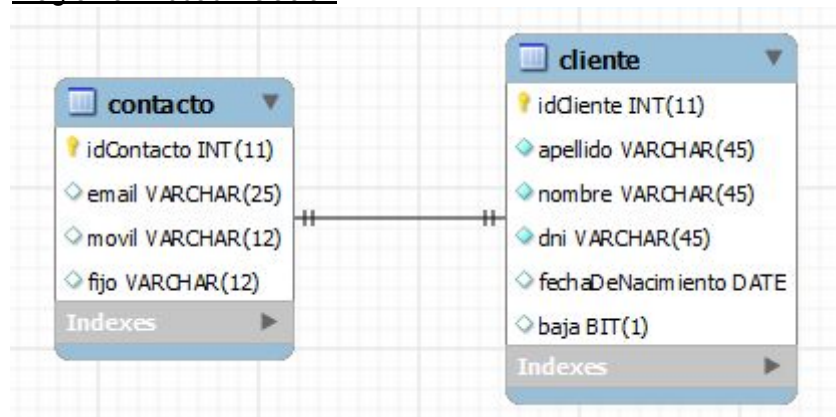
Bibliografía: ver programa Hibernate

Proyecto: Hibernate-Uno-a-Uno

1. Importar bd-hibernate-uno-a-uno.sql

La clave primaria de la entidad contacto va depender de cliente y para cada cliente existe a lo sumo un contacto.

Diagrama Entidad Relación:

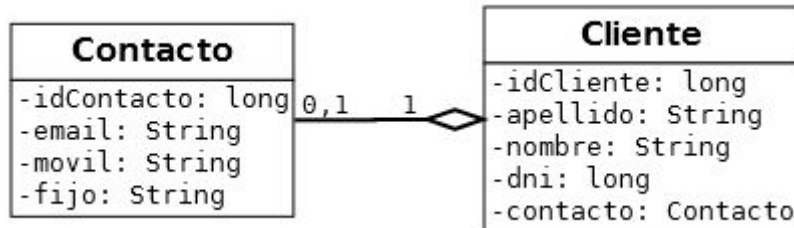


Cuando se agrega un registro cliente, en el caso de agregar el contacto el idContacto sera el mismo número de idCliente

Foreign Key Name		Referenced Table	Column		Referenced Column
fk_cliente		`bd-hibernate-uno-a-uno`.`cliente`	<input checked="" type="checkbox"/> idContacto		idCliente
			<input type="checkbox"/> email		
			<input type="checkbox"/> movil		
			<input type="checkbox"/> fijo		

## Diagrama de clases

Ahora contacto va ser atributo de Cliente, ya que cuando traemos un objeto Cliente también nos interesa saber los datos de contacto.



```
package datos;
```

```
import java.util.GregorianCalendar;
import funciones.Funciones;
```

```
public class Cliente {
    private long idCliente;
    private String apellido;
    private String nombre;
    private int dni;
    private GregorianCalendar fechaDeNacimiento;
    private boolean baja;
    private Contacto contacto;

    public Cliente(){}

    public Cliente(String apellido, String nombre, int dni,
        GregorianCalendar fechaDeNacimiento) {
        super();
        this.apellido = apellido;
        this.nombre = nombre;
        this.dni= dni;
        this.fechaDeNacimiento = fechaDeNacimiento;
        this.baja=false;
    }

    public long getIdCliente() {
        return idCliente;
    }

    protected void setIdCliente(long idCliente) {
        this.idCliente = idCliente;
    }

    public String getApellido() {
        return apellido;
    }

    public void setApellido(String apellido) {
        this.apellido = apellido;
    }
}
```

```

public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public int getDni() {
    return dni;
}

public void setDni(int dni) {
    this.dni = dni;
}

public GregorianCalendar getFechaDeNacimiento() {
    return fechaDeNacimiento;
}

public void setFechaDeNacimiento(GregorianCalendar fechaDeNacimiento) {
    this.fechaDeNacimiento = fechaDeNacimiento;
}

public boolean isBaja() {
    return baja;
}

public void setBaja(boolean baja) {
    this.baja = baja;
}

public Contacto getContacto() {
    return contacto;
}

public void setContacto(Contacto contacto) {
    this.contacto = contacto;
}

public String toString(){
    return (idCliente+" "+apellido+" "+nombre+" DNI: "+dni+" F.de Nacimiento:
"+Funciones.traerFechaCorta(fechaDeNacimiento)+" "+baja);
}
}

```

1)

```

package datos;

public class Contacto {
    private long idContacto;
    private String email;
    private String movil;
    private String fijo;
    private Cliente cliente;

    public Contacto(){}

    public Contacto(String email, String movil, String fijo, Cliente cliente) {
        this.email = email;
        this.movil = movil;
        this.fijo = fijo;
        this.cliente=cliente;
    }

    public long getIdContacto() {
        return idContacto;
    }
    protected void setIdContacto(long idContacto) {
        this.idContacto = idContacto;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getMovil() {
        return movil;
    }
    public void setMovil(String movil) {
        this.movil = movil;
    }
    public String getFijo() {
        return fijo;
    }
    public void setFijo(String fijo) {
        this.fijo = fijo;
    }
    public Cliente getCliente() {
        return cliente;
    }
    public void setCliente(Cliente cliente) {
        this.cliente = cliente;
    }

    public String toString(){
        return email+" "
        +" Movil: "+ movil
        +" Teléfono: "+ fijo;
    }
}

```

Mapeos de configuración en Cliente.hbm.xml agregamos la relación uno-a-uno

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>

<class name="datos.Cliente" table="cliente">
  <id column="idCliente" name="idCliente">
    <generator class="identity"/>
  </id>
  <property column="apellido" name="apellido" type="string"/>
  <property column="nombre" name="nombre" type="string"/>
  <property column="dni" name="dni" type="int"/>
  <property column="fechaDeNacimiento" name="fechaDeNacimiento" type="calendar"/>
  <property column="baja" name="baja" type="boolean"/>

  <one-to-one name="contacto" class="datos.Contacto" ></one-to-one>
</class>
</hibernate-mapping>

```

Creamos en el paquete mapeos Contacto.hbm.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>

  <class name="datos.Contacto" table="contacto" >
    <id name="idContacto" column="idContacto">
      <generator class="foreign">
        <param name="property">cliente</param>
      </generator>
    </id>
    <property name="email" type="string" column="email" />
    <property name="movil" type="string" column="movil" />
    <property name="fijo" type="string" column="fijo" />

    <one-to-one name="cliente" class="datos.Cliente" ></one-to-one>
  </class>

</hibernate-mapping>

```

Creamos en el ContactoDao quedan para implementar los métodos modificar y eliminar.

```
package dao;

import org.hibernate.HibernateException;
import org.hibernate.Session;
import org.hibernate.Transaction;
import datos.Contacto;

public class ContactoDao {
    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 agregar(Contacto 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;
    }
}
```

```
package dao;

import java.util.List;

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 agregar(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 actualizar(Cliente objeto) throws HibernateException {
        try {
            iniciaOperacion();
            session.update(objeto);
            tx.commit();
        } catch (HibernateException he) {
            manejaExcepcion(he);
            throw he;
        } finally {
            session.close();
        }
    }

    public void eliminar(Cliente objeto) throws HibernateException {
        try {
            iniciaOperacion();
            session.delete(objeto);
            tx.commit();
        } catch (HibernateException he) {
            manejaExcepcion(he);
            throw he;
        } finally {
            session.close();
        }
    }

    public Cliente traerCliente(long idCliente) throws HibernateException {
        Cliente objeto = null;
        try {
            iniciaOperacion();
            objeto = (Cliente) session.get(Cliente.class, idCliente);
        } finally {
            session.close();
        }
        return objeto;
    }
}

```

```

    public Cliente traerCliente(int 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> traerCliente() 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;
    }

    public Cliente traerClienteYContacto(long idCliente) throws HibernateException {
        Cliente objeto = null;
        try {
            iniciaOperacion();
            String hql="from Cliente c inner join fetch c.contacto where c.idCliente ="
+ idCliente;
            objeto = (Cliente) session.createQuery(hql).uniqueResult();
        } finally {
            session.close();
        }
        return objeto;
    }
}

```



TEST:

```
package test;

import dao.ClienteDao;
import dao.ContactoDao;
import datos.Cliente;
import datos.Contacto;

public class TestAgregarContacto {
    public static void main(String[] args) {

        ClienteDao dao=new ClienteDao();
        long idCliente=1;
        Cliente c=dao.traerCliente(idCliente);
        System.out.println(c);
        Contacto con=new
Contacto("ajaramillo@unla.edu.ar","11-1111-1111","011-1111-1111", c);
        ContactoDao dao1=new ContactoDao();
        dao1.agregar(con);

    }
}
```

```
package test;

import dao.ClienteDao;
import datos.Cliente;

public class TestTraerClienteYContacto {
    public static void main(String[] args) {

        ClienteDao dao=new ClienteDao();
        long idCliente=1;
        Cliente c=dao.traerClienteYContacto(idCliente);
        System.out.println("\ntraer Cliente y contacto\n"+
c+"\n"+c.getContacto());

    }
}
```