

IDE: Eclipse

Persistencia de datos: MySQL

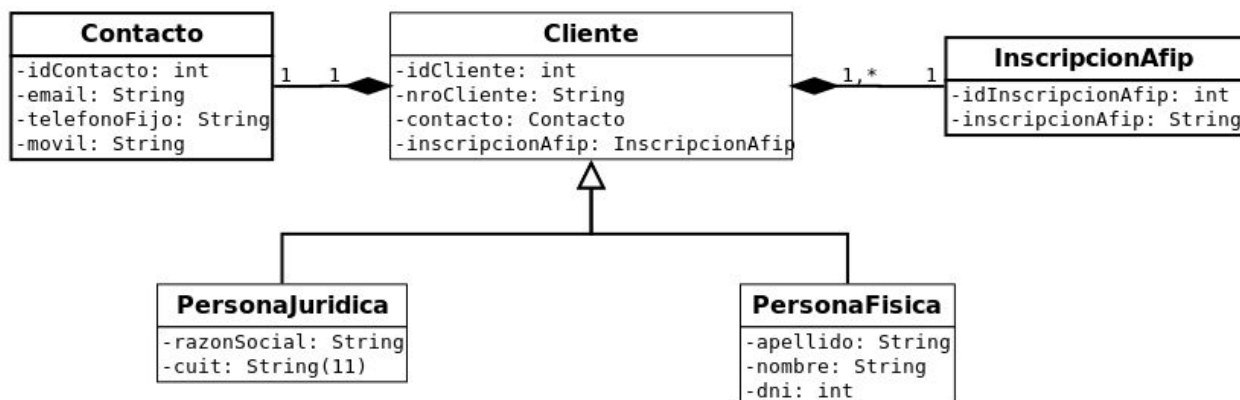
Bibliografía: ver programa Hibernate

Framework Hibernate Herencia

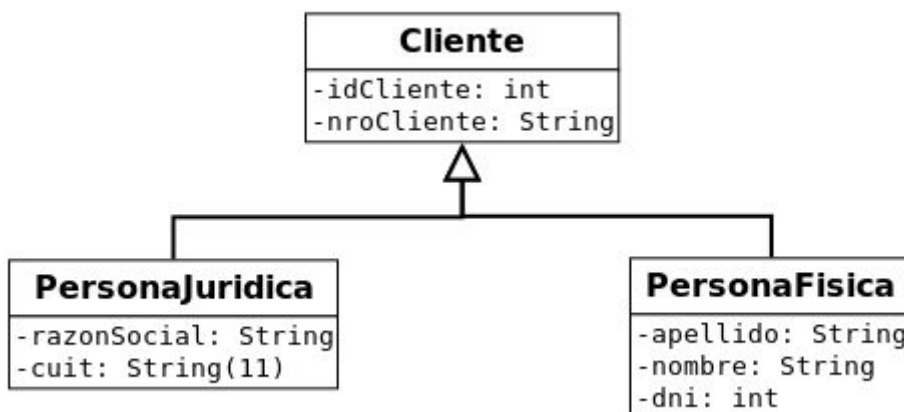
Para ilustrar herencia planteamos un problema real para clientes de una empresa de telefonía o de un servicio público o un banco, etc.

Cliente encapsula, nroCliente (puede ser un número con un dígito verificador), inscripcionAfip y contacto determinando una relación de composición ("tiene un"). PersonaJuridica (empresas, universidad, cooperativas, etc) y PersonaFisica (ciudadano) heredan ("es un/una") de Cliente.

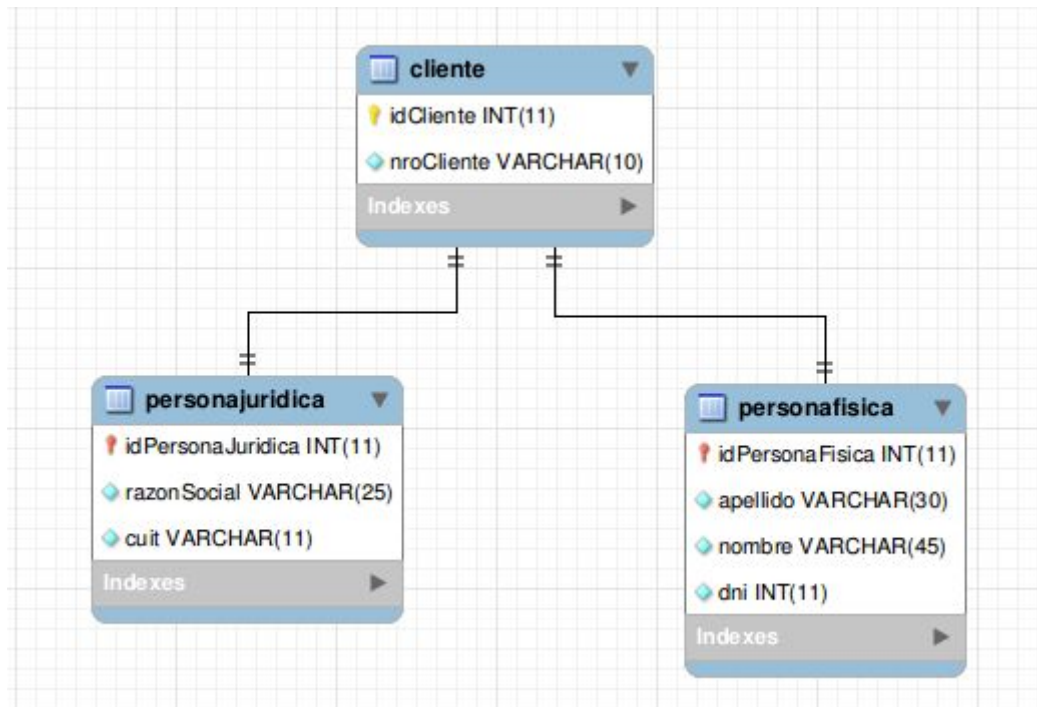
UML Diagrama de Clases - Capa de Datos



Para simplificar vamos a implementar solo la relación de herencia, ya que las relaciones de composición se describieron en guías previas (Hibernate una-a-muchos y Hibernate uno-a-uno).



Modelo entidad relación: Database: bd-hibernate-herencia



Modelo entidad relación Script:

-- MySQL dump 10.13 Distrib 5.7.18, for Linux (x86_64)

--

-- Host: localhost Database: bd-hibernate-herencia

-- Server version 5.7.18-0ubuntu0.16.04.1

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

--

-- Table structure for table `cliente`

--

```
DROP TABLE IF EXISTS `cliente`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `cliente` (
  `idCliente` int(11) NOT NULL AUTO_INCREMENT,
  `nroCliente` varchar(10) NOT NULL,
  PRIMARY KEY (`idCliente`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
```

```

/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `cliente`
--

LOCK TABLES `cliente` WRITE;
/*!40000 ALTER TABLE `cliente` DISABLE KEYS */;
INSERT INTO `cliente` VALUES (1,'11111111'),(2,'22222222');
/*!40000 ALTER TABLE `cliente` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `personafisica`
--

DROP TABLE IF EXISTS `personafisica`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `personafisica` (
  `idPersonaFisica` int(11) NOT NULL,
  `apellido` varchar(30) NOT NULL,
  `nombre` varchar(45) NOT NULL,
  `dni` int(11) NOT NULL,
  PRIMARY KEY (`idPersonaFisica`),
  CONSTRAINT `fk_personafisica_1` FOREIGN KEY (`idPersonaFisica`) REFERENCES `cliente` (`idCliente`) ON
DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `personafisica`
--

LOCK TABLES `personafisica` WRITE;
/*!40000 ALTER TABLE `personafisica` DISABLE KEYS */;
INSERT INTO `personafisica` VALUES (1,'Lopez','Juan',11111111);
/*!40000 ALTER TABLE `personafisica` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `personajuridica`
--

DROP TABLE IF EXISTS `personajuridica`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `personajuridica` (
  `idPersonaJuridica` int(11) NOT NULL,
  `razonSocial` varchar(25) NOT NULL,
  `cuit` varchar(11) NOT NULL,
  PRIMARY KEY (`idPersonaJuridica`),
  CONSTRAINT `fk_personajuridica_1` FOREIGN KEY (`idPersonaJuridica`) REFERENCES `cliente` (`idCliente`) ON
DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--

```

```
-- Dumping data for table `personajuridica`  
--
```

```
LOCK TABLES `personajuridica` WRITE;  
/*!40000 ALTER TABLE `personajuridica` DISABLE KEYS */;  
INSERT INTO `personajuridica` VALUES (2,'UNLa','300000001');  
/*!40000 ALTER TABLE `personajuridica` ENABLE KEYS */;  
UNLOCK TABLES;  
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;  
  
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;  
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;  
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;  
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

```
-- Dump completed
```

Capa de datos:

package datos;

```
public class Cliente {  
    private int idCliente;  
    private String nroCliente;  
  
    public Cliente(){}  
  
    public Cliente(String nroCliente) {  
        super();  
        this.nroCliente = nroCliente;  
    }  
  
    public int getIdCliente() {  
        return idCliente;  
    }  
  
    protected void setIdCliente(int idCliente) {  
        this.idCliente = idCliente;  
    }  
  
    public String getNroCliente() {  
        return nroCliente;  
    }  
  
    public void setNroCliente(String nroCliente) {  
        this.nroCliente = nroCliente;  
    }  
  
    @Override  
    public String toString() {  
        return "Cliente [idCliente=" + idCliente + ", nroCliente=" + nroCliente + "];"  
    }  
}
```

```
package datos;
```

```
public class PersonaJuridica extends Cliente{
    private int idPersonaJuridica;
    private String razonSocial;
    private String cuit;

    public PersonaJuridica() {}
    public PersonaJuridica(String nroCliente, String razonSocial, String cuit) {
        super(nroCliente);
        this.razonSocial = razonSocial;
        this.cuit = cuit;
    }

    public int getIdPersonaJuridica() {
        return idPersonaJuridica;
    }
    public void setIdPersonaJuridica(int idPersonaJuridica) {
        this.idPersonaJuridica = idPersonaJuridica;
    }
    public String getRazonSocial() {
        return razonSocial;
    }
    public void setRazonSocial(String razonSocial) {
        this.razonSocial = razonSocial;
    }
    public String getCuit() {
        return cuit;
    }
    public void setCuit(String cuit) {
        this.cuit = cuit;
    }
    @Override
    public String toString() {
        return "PersonaJuridica [idPersonaJuridica=" + idPersonaJuridica + ", nroCliente=" +
this.getNroCliente()+", razonSocial=" + razonSocial + ", cuit=" + cuit + "];"
    }
}

}
```

```
package datos;
```

```
public class PersonaFisica extends Cliente{
    private int idPersonaFisica;
    private String apellido;
    private String nombre;
    private int dni;

    public PersonaFisica() {}
    public PersonaFisica(String nroCliente, String apellido, String nombre, int dni) {
        super(nroCliente);
        this.apellido = apellido;
        this.nombre = nombre;
        this.dni = dni;
    }

    public int getIdPersonaFisica() {
        return idPersonaFisica;
    }
}
```

```

    }
    protected void setIdPersonaFisica(int idPersonaFisica) {
        this.idPersonaFisica = idPersonaFisica;
    }
    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;
    }
    @Override
    public String toString() {
        return "PersonaFisica [idPersonaFisica=" + idPersonaFisica + ", nroCliente=" + this.getNroCliente() + ",
apellido=" + apellido + ", nombre=" + nombre
        + ", dni=" + dni + "]";
    }
}

```

Archivos de mapeo, Cliente.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.Cliente" table="cliente">
    <id column="idCliente" name="idCliente">
        <generator class="identity"/>
    </id>
    <property name="nroCliente" column="nroCliente" type="string"/>

    <joined-subclass name="datos.PersonaJuridica" table="personajuridica">
        <key column="idPersonaJuridica" />
        <property column="razonSocial" name="razonSocial" type="string"/>
        <property column="cuit" name="cuit" type="string"/>
    </joined-subclass>

    <joined-subclass name="datos.PersonaFisica" table="personafisica">
        <key column="idPersonaFisica" />
        <property column="apellido" name="apellido" type="string"/>
        <property column="nombre" name="nombre" type="string"/>
        <property column="dni" name="dni" type="int"/>
    </joined-subclass>
</class>
</hibernate-mapping>

```

Capa de persistencia

```
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 static ClienteDao instancia=null; // Patrón Singleton

    protected ClienteDao() {}

    public static ClienteDao getInstance(){
        if(instancia==null)
            instancia=new ClienteDao();
        return instancia;
    }

    protected void iniciarOperacion() throws HibernateException {
        session = HibernateUtil.getSessionFactory().openSession();
        tx = session.beginTransaction();
    }

    protected void manejarExcepcion(HibernateException he) throws HibernateException {
        tx.rollback();
        throw new HibernateException("ERROR en la capa de acceso a datos", he);
    }

    public Cliente traer(int idCliente){
        Cliente objeto = null;
        try {
            iniciarOperacion();
            objeto = (Cliente) session.createQuery("from Cliente c where c.idCliente =" + idCliente).uniqueResult();
        } finally {
            session.close();
        }
        return objeto;
    }

    @SuppressWarnings("unchecked")
    public List<Cliente> traer() throws HibernateException {
        List<Cliente> lista=null;
        try {
            iniciarOperacion();
            lista=session.createQuery("from Cliente").list();
        } finally {
            session.close();
        }
        return lista;
    }
}
```

Capa de negocio

```
package negocio;

import java.util.List;

import dao.ClienteDao;
import datos.Cliente;

public class ClienteAbm {
    private static ClienteAbm instancia=null; // Patrón Singleton

    protected ClienteAbm() {}

    public static ClienteAbm getInstance(){
        if(instancia==null)
            instancia=new ClienteAbm();
        return instancia;
    }

    public Cliente traer(int idCliente){
        return ClienteDao.getInstance().traer(idCliente);
    }

    public List<Cliente> traer(){
        return ClienteDao.getInstance().traer();
    }
}
```

Test por consola:

```
package test;

import datos.Cliente;
import negocio.ClienteAbm;

public class TestCliente {
    public static void main(String[] args) {
        int idCliente=1;
        System.out.println("+ traer("+idCliente+")");
        System.out.println(ClienteAbm.getInstance().traer(idCliente));

        idCliente=2;
        System.out.println("\n+ traer("+idCliente+")");
        System.out.println(ClienteAbm.getInstance().traer(idCliente));

        System.out.println("\n+ traer()");
        for (Cliente c: ClienteAbm.getInstance().traer()) {
            System.out.println(c);
        }
    }
}
```


Run:

+ traer(1)

PersonaFisica [idPersonaFisica=0, nroCliente=11111111, apellido=Lopez, nombre=Juan, dni=11111111]

+ traer(2)

PersonaJuridica [idPersonaJuridica=0, nroCliente=22222222, razonSocial=UNLa, cuit=300000001]

+ traer()

PersonaJuridica [idPersonaJuridica=0, nroCliente=22222222, razonSocial=UNLa, cuit=300000001]

PersonaFisica [idPersonaFisica=0, nroCliente=11111111, apellido=Lopez, nombre=Juan, dni=11111111]