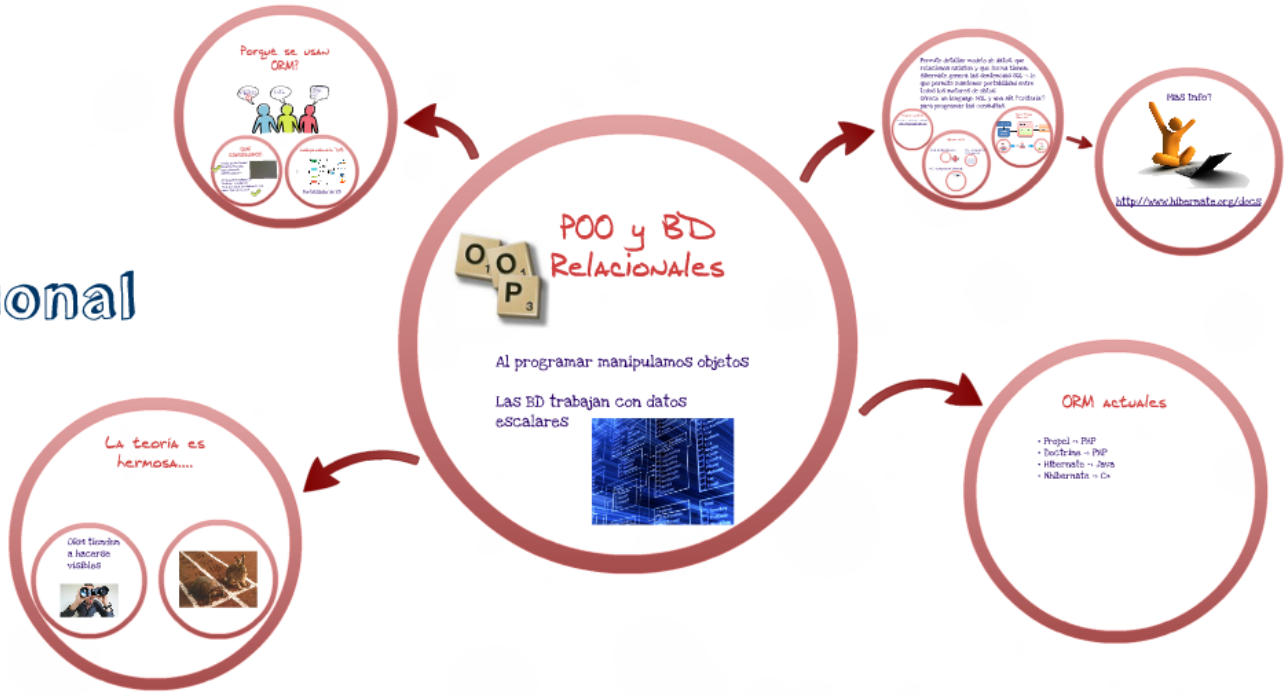
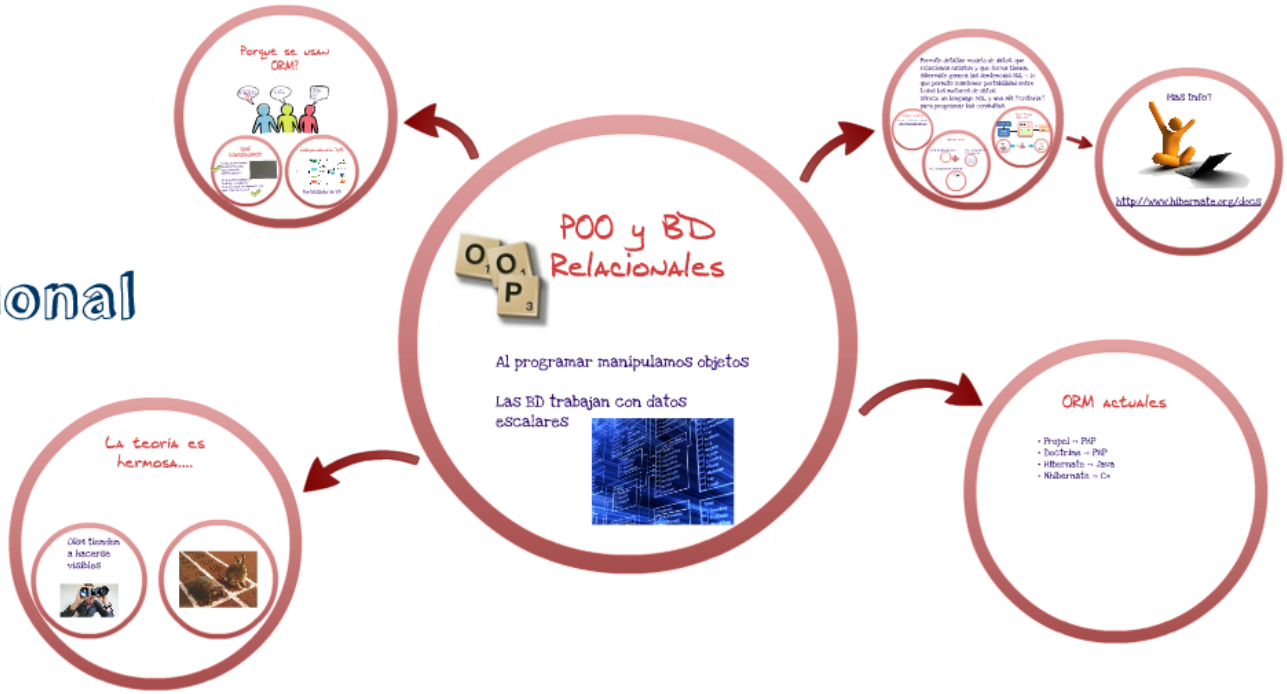
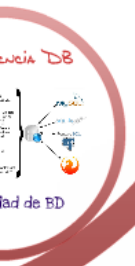
ORM - Object Relational Mapping



ORM - Object Relational Mapping



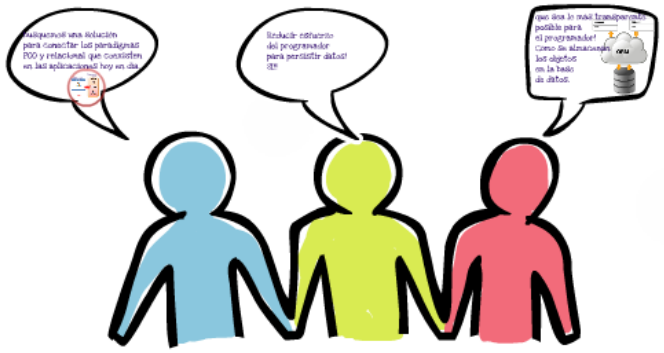


POO y BD Relacionales

Al programar manipulamos objetos

Las BD trabajan con datos escalares

Porqué se USAN ORM?

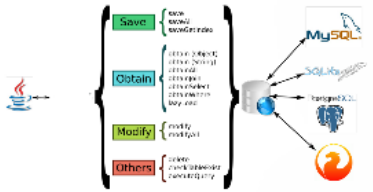


QUE CONSEGUIMOS!

mejora productividad
aumento facilidad mantenimiento
INDEPENDENCIA Y

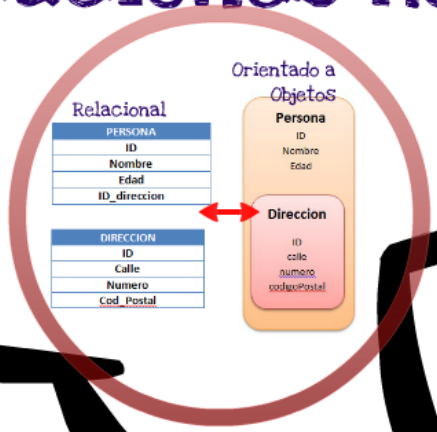
ABSTRACCIÓN sistema BD
Fácil implementación
Fácil ejecución pruebas unitarias
corrección de código!!

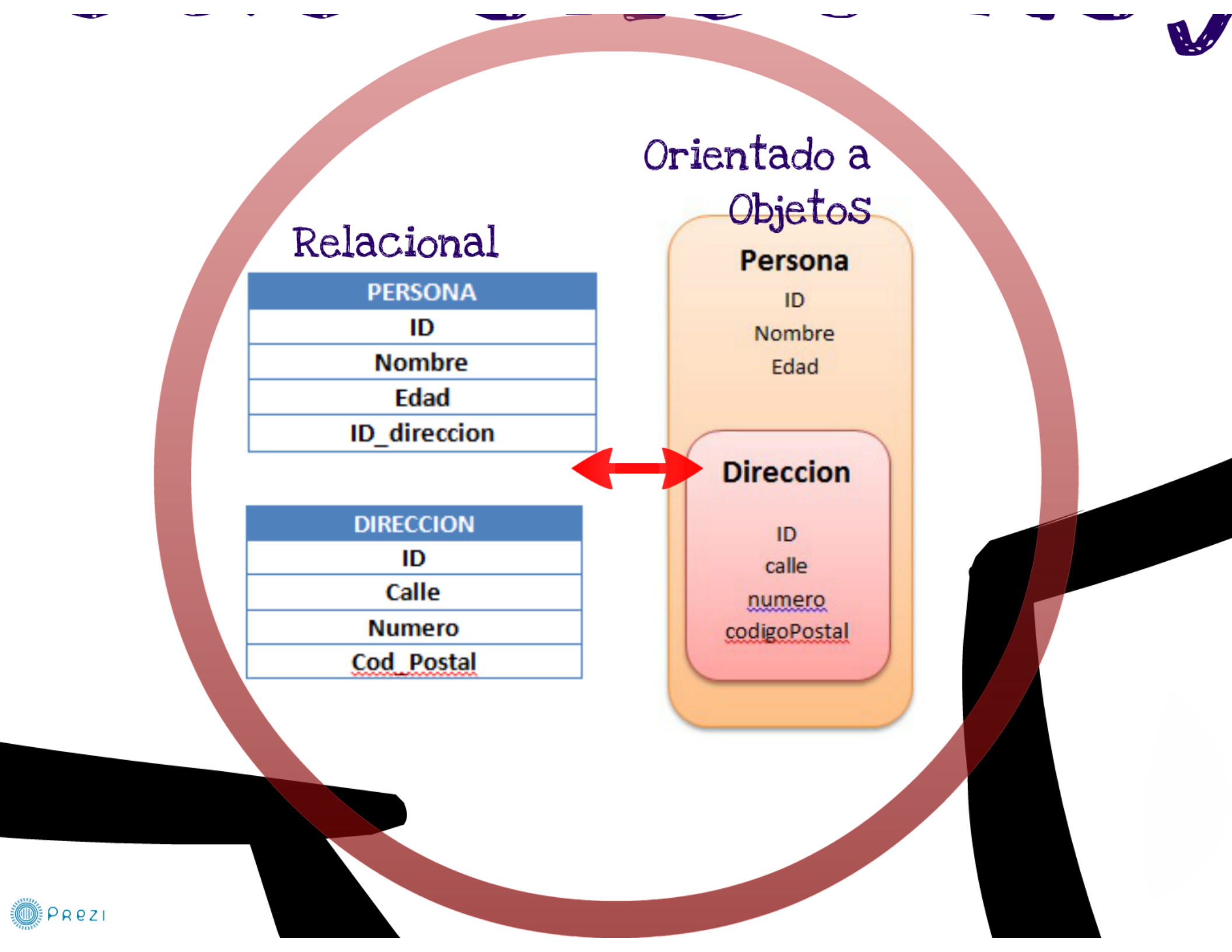
Independencia DB



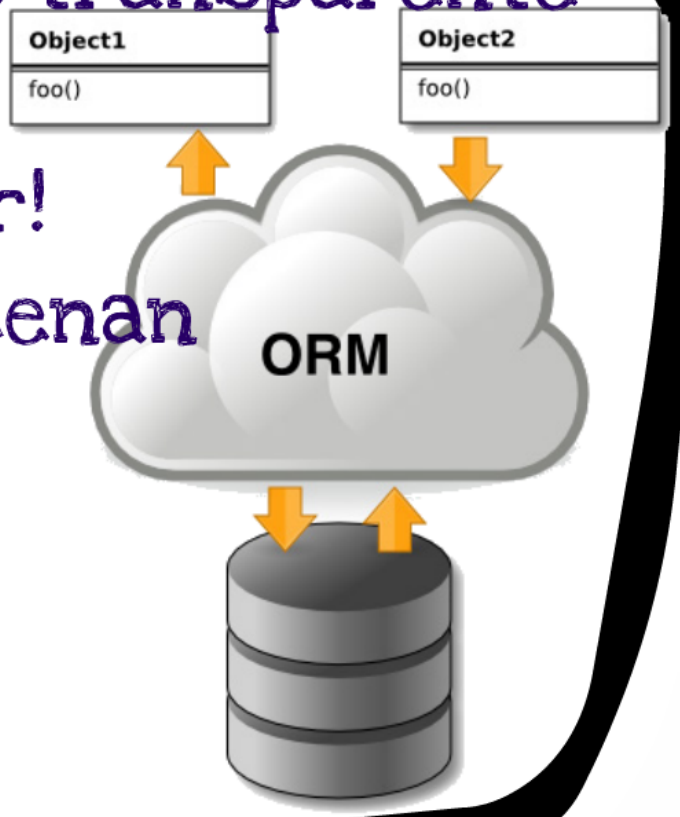
Portabilidad de BD

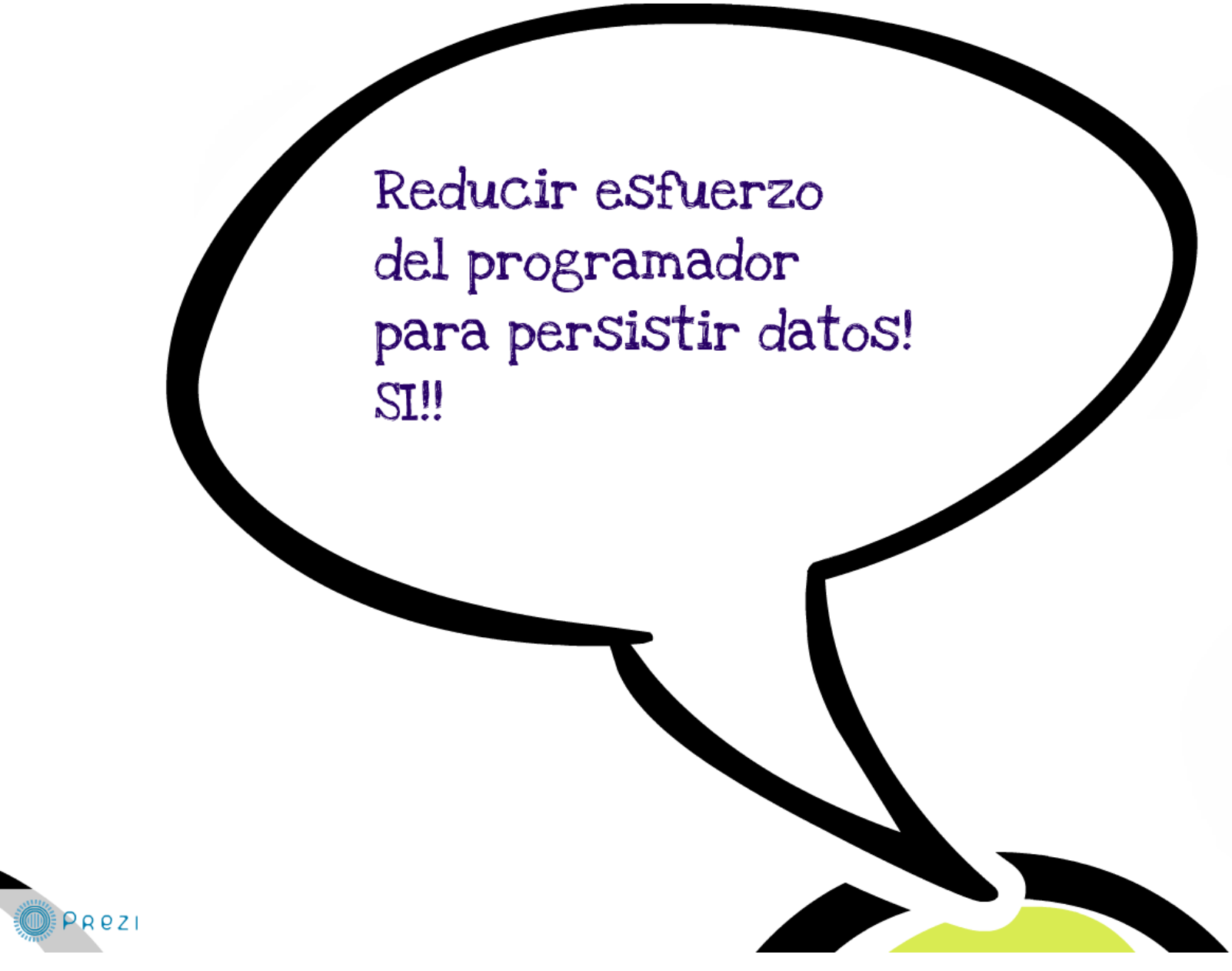
busquemos una solución para conectar los paradigmas POO y relacional que coexisten en las aplicaciones hoy en día.




Orientado a Objetos
Relacional
PERSONA
ID
Nombre
Edad
ID_direccion
DIRECCION
ID
Calle
Numero
Cod_Postal
Persona
ID
Nombre
Edad
Direccion
ID
calle
numero
codigoPostal

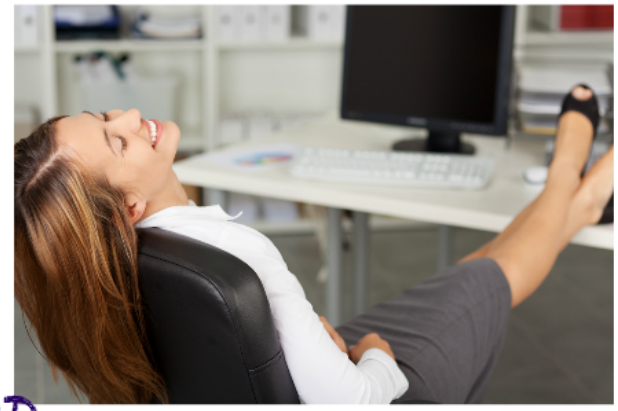
que sea lo más transparente
posible para
el programador!
Cómo se almacenan
los objetos
en la base
de datos.





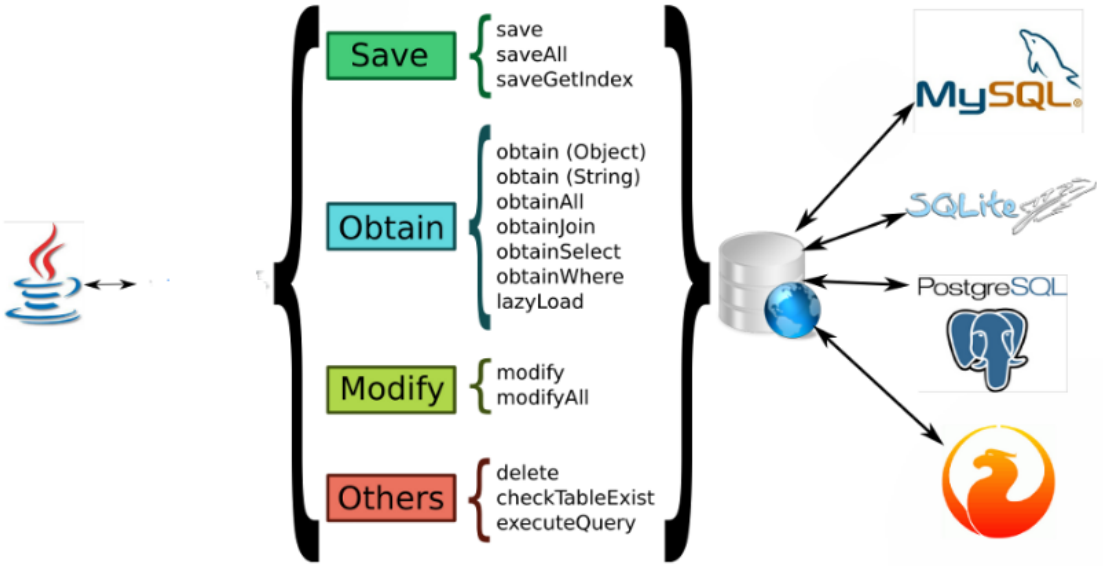
QUE CONSEGUIMOS!

mejora productividad
aumento facilidad
mantenimiento
INDEPENDENCIA Y



ABSTRACCIÓN sistema BD
Fácil implementación
Fácil ejecución pruebas unitarias
corrección de código!!

Independencia DB



Portabilidad de BD

La teoría es hermosa....

ORM tienden a hacerse visibles

ORM tienden a hacerse visibles

ORM actuales

- Propel -> PHP
- Doctrine -> PHP
- Hibernate -> Java
- Nhibernate -> C#

HIBERNATE

Permite detallar modelo de datos, que relaciones existen y que forma tienen.
Hibernate genera las sentencias SQL -> lo que permite mantener portabilidad entre todos los motores de datos.
Ofrece un lenguaje: HQL y una API ("criteria") para programar las consultas.

Preparar ambiente

Descargar el zip con las clases de hibernate

http://www.hibernate.org

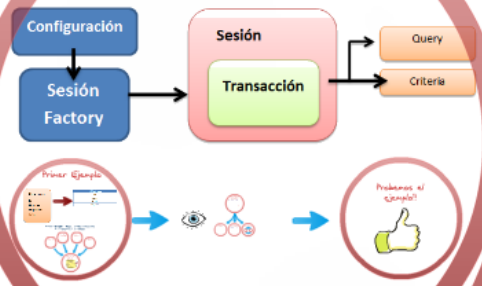
Ejemplo simple

clase de Java (Persona)

xml - mapeo para la clase Persona

xml - configuración hibernate

Cómo Trabaja Hibernate?



Preparar ambiente

Descargar el zip con las clases de hibernate

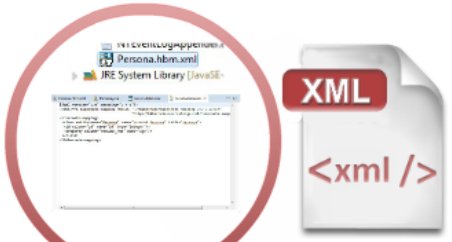
http://www.hibernate.org

clase de

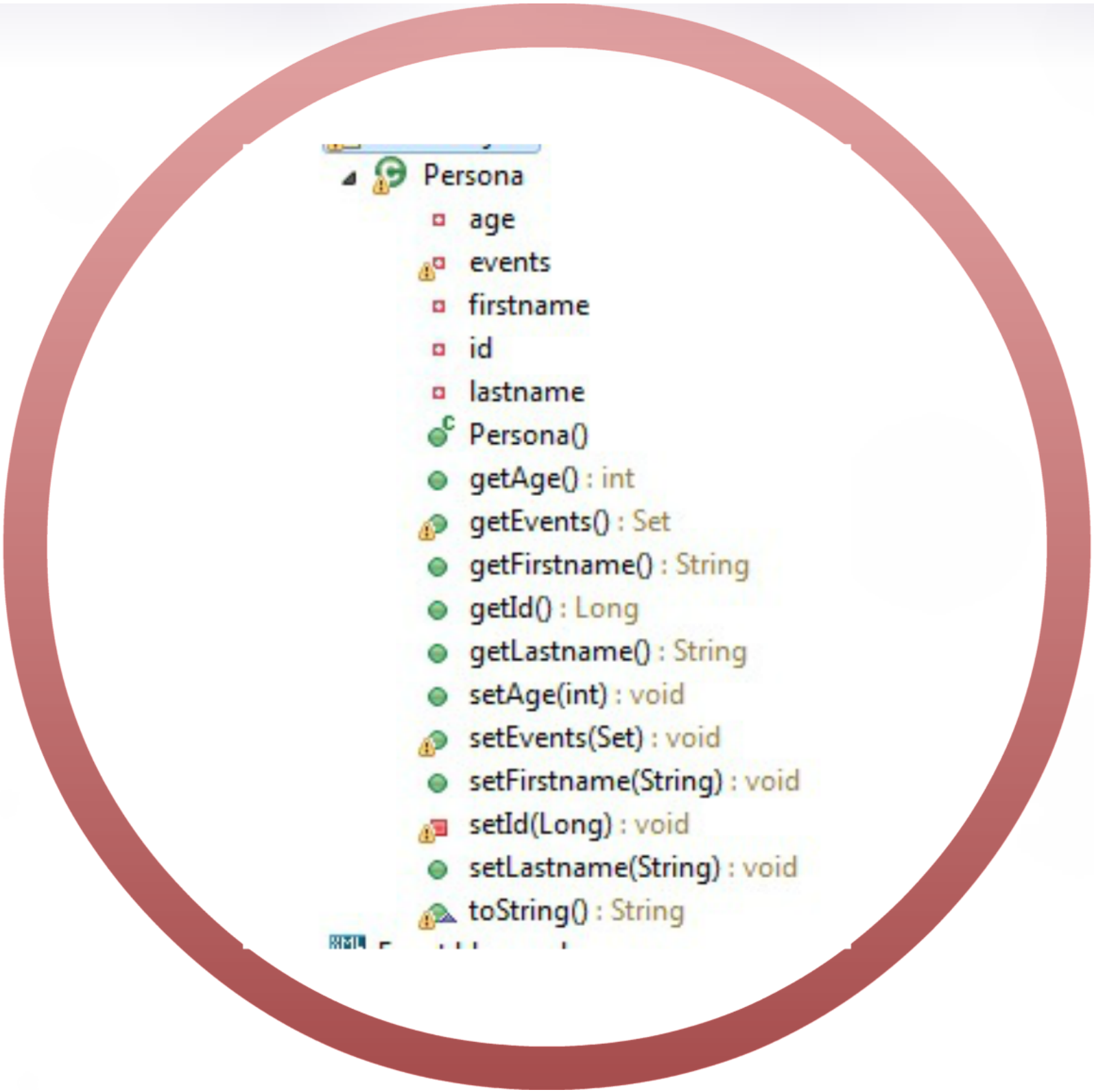
Ejemplo simple

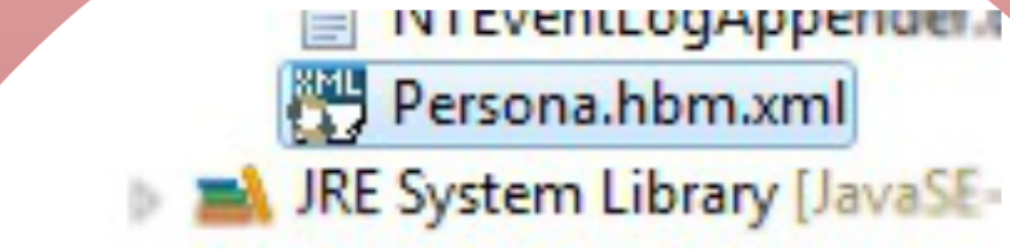
clase de Java (Persona)

xml - mapeo para la clase Persona



xml - configuración hibernate

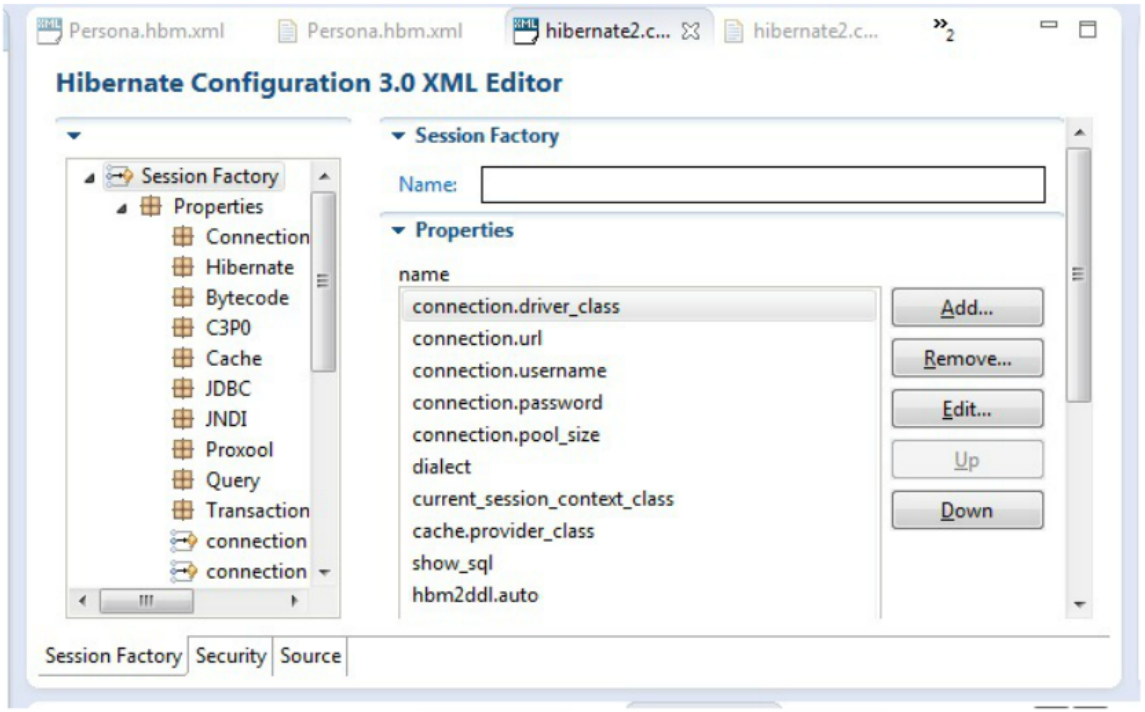

Persona
age
events
firstname
id
lastname
Persona()
getAge() : int
getEvents() : Set
getFirstname() : String
getId() : Long
getLastname() : String
setAge(int) : void
setEvents(Set) : void
setFirstname(String) : void
setId(Long) : void
setLastname(String) : void
toString() : String



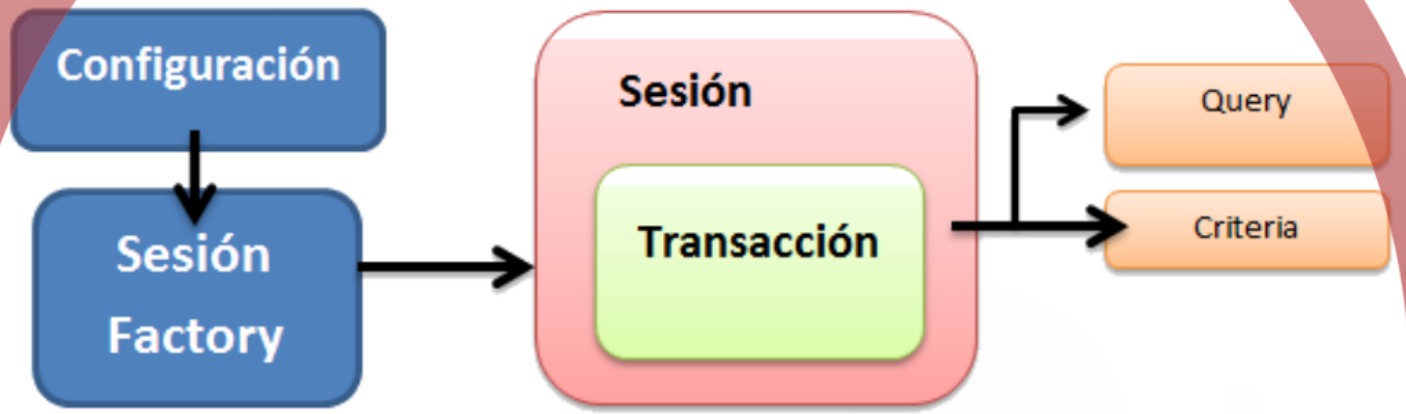
Persona.hbm.xml | Persona.java | Persona.hbm.xml | Persona.hbm.xml

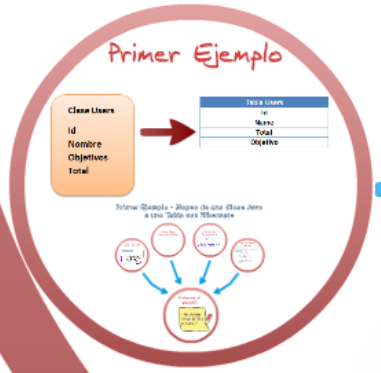
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
                                   "http://hibernate.sourceforge.net/hibernate-mapp
<hibernate-mapping>
 <class entity-name="Persona" name="prueba1.Persona" table="persona">
  <id column="id" name="id" type="Integer"/>
  <property column="PERSONA_AGE" name="age"/>
 </class>
</hibernate-mapping>
```

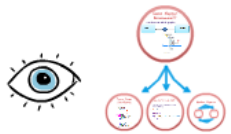

hibernate2.cfg.xml


Persona.hbm.xml
Persona.hbm.xml
hibernate2.c...
hibernate2.c...
Hibernate Configuration 3.0 XML Editor
Session Factory
Properties
Connection
Hibernate
Bytecode
C3P0
Cache
JDBC
JNDI
Proxool
Query
Transaction
connection
connection
Session Factory
Name:
Properties
name
connection.driver_class
connection.url
connection.username
connection.password
connection.pool_size
dialect
current_session_context_class
cache.provider_class
show_sql
hbm2ddl.auto
Add...
Remove...
Edit...
Up
Down
Session Factory
Security
Source

Cómo Trabaja Hibernate?







Sesión Factory

Transacción

Query

Criteria

Primer Ejemplo

Clase Users

Id
Nombre
Objetivos
Total

Tabla Users
Id
Name
Total
Objetivo

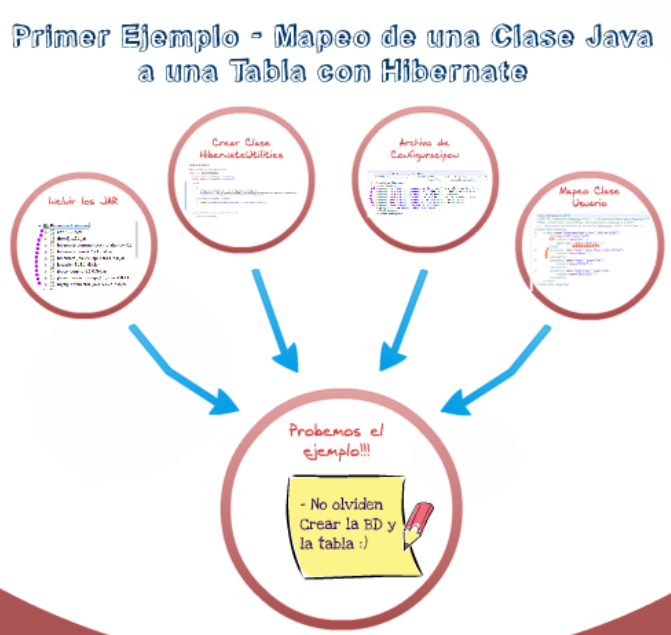
Probemos el ejemplo!!

Primer Ejemplo

Clase Users

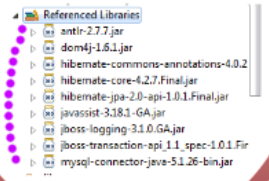
Id
Nombre
Objetivos
Total

Tabla Users
Id
Name
Total
Objetivo



Primer Ejemplo - Mapeo de una Clase Java a una Tabla con Hibernate

Incluir los JAR



Crear Clase HibernateUtilities

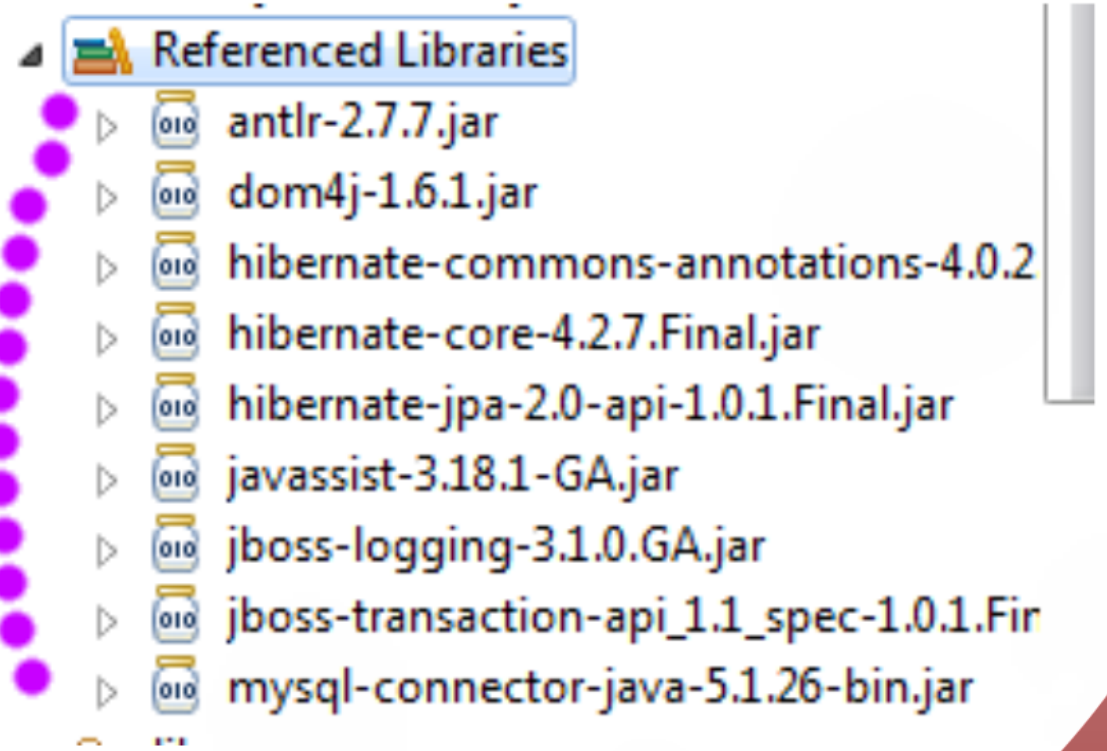
Archivo de Configuracipon

Mapeo Clase Usuario

Probemos el ejemplo!!!

- No olviden Crear la BD y la tabla :)

Incluir los JAR



Crear Clase HibernateUtilities

```java
package programaSimple;

import org.hibernate.HibernateException;

public class HibernateUtilities {

    private static SessionFactory sessionFactory;
    private static ServiceRegistry serviceRegistry;

    static
    {
        try
        {
            Configuration configuration = new Configuration().configure();
            serviceRegistry = new ServiceRegistryBuilder().applySettings(configuration.getProperties()).buildServiceI
            sessionFactory = configuration.buildSessionFactory(serviceRegistry);
        }
        catch (HibernateException exception)
        {
            System.out.println("problema con hibernate, al crear session factory");
        }

    }

    public static SessionFactory getSessionFactory()
    {
        return sessionFactory;
    }
}
```

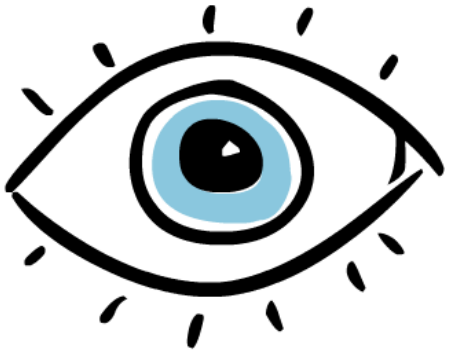
Archivo de Configuracipon

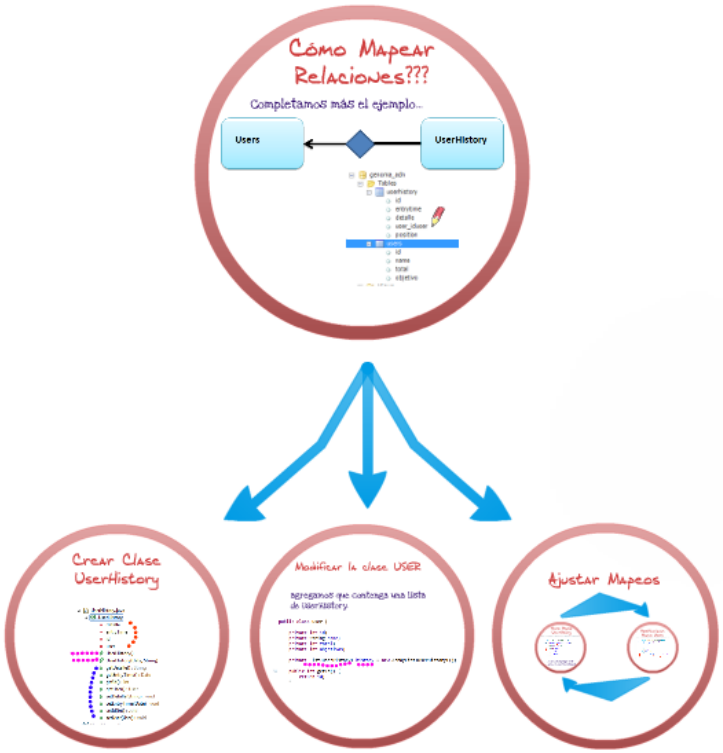
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0/
                                         "http://hibernate.sourceforge.net/hibernate-cc
<hibernate-configuration>
 <session-factory>
  <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
  <property name="hibernate.connection.url">jdbc:mysql://localhost:3306</property>
  <property name="hibernate.default_schema">genoma_adn</property>
  <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
  <property name="hibernate.connection.username">hibernate</property>
  <property name="hibernate.connection.password">hibernate</property>
  <property name="hibernate.show_sql">true</property>
  <mapping resource="programaSimple/User.hbm.xml"/>
 </session-factory>
</hibernate-configuration>
```

Mapeo Clase Usuario

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!-- Generated 20/11/2013 12:53:48 by Hibernate Tools 3.4.0.CR1 -->
<hibernate-mapping>
    <class name="programaSimple.User" table="USERS">
        <id name="id" type="int">
            <column name="ID" />
            <generator class="identity" />
        </id>
        <property name="name" type="java.lang.String">
            <column name="NAME" />
        </property>
        <property name="total" type="int">
            <column name="TOTAL" />
        </property>
        <property name="objetivos" type="int">
            <column name="OBJETIVO" />
        </property>
    </class>
</hibernate-mapping>
```

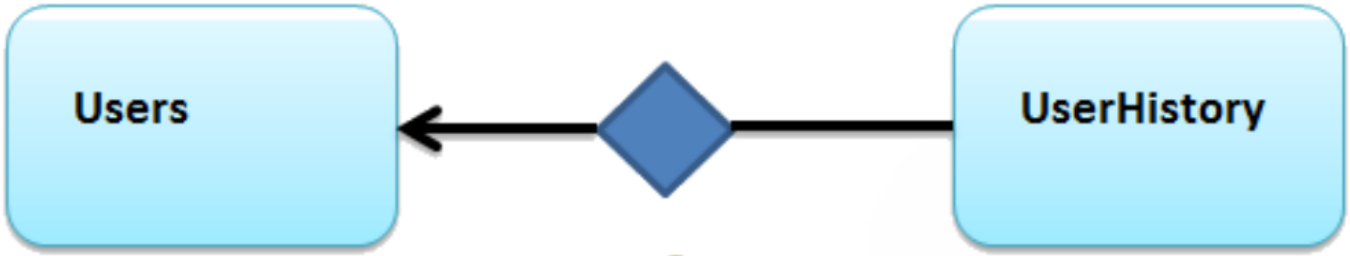
Probemos el ejemplo!!!

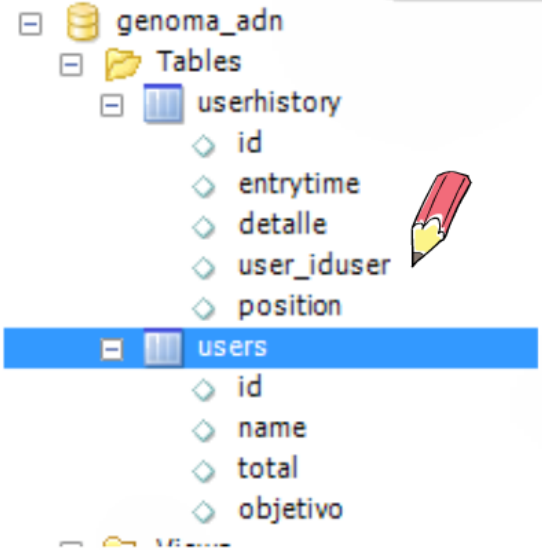



Cómo Mapear Relaciones???
Completamos más el ejemplo...
Users
UserHistory
Crear Clase UserHistory
Modificar la clase USER
Ajustar Mapeos

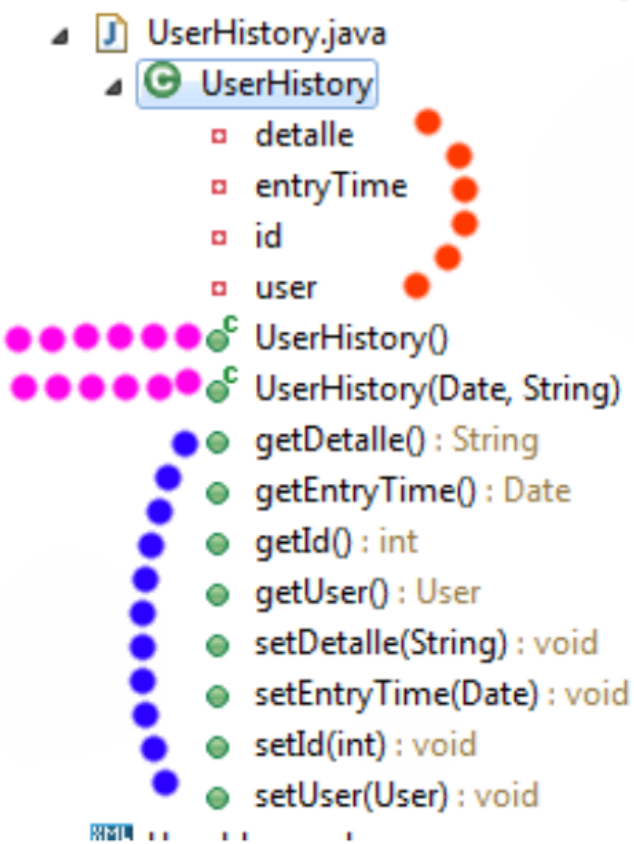
Cómo Mapear Relaciones???

Completamos más el ejemplo...





Crear Clase UserHistory



Modificar la clase USER

agregamos que contenga una lista de UserHistory.

```
public class User {

    private int id;
    private String name;
    private int total;
    private int objetivos;

    private List<UserHistory> history = new ArrayList<UserHistory>();

    public int getId() {
        return id;
    }
```

Ajustar Mapeos

Nuevo Mapeo UserHistory

No Olviden agregar este mapeo al archivo cfg.xml!!

Modificaciones Mapeo Users

Nuevo Mapeo UserHistory

User.java | *UserHistory.java | UserHistory.hbm.xml | User.hbm.xml | hibernate.cfg.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!-- Generated 20/11/2013 19:29:46 by Hibernate Tools 3.4.0.CR1 -->
<hibernate-mapping>
    <class name="programaSimple.UserHistory" table="USERHISTORY">
        <id name="id" type="int">
            <column name="ID" />
            <generator class="increment" />
        </id>
        <many-to-one name="user" class="programaSimple.User" not-null="true">
            <column name="USER_IDUSER" />
        </many-to-one>
        <property name="entryTime" type="java.util.Date">
            <column name="ENTRYTIME" />
        </property>
        <property name="detalle" type="java.lang.String">
            <column name="DETALLE" />
        </property>
    </class>
</hibernate-mapping>
```

No Olviden agregar este mapeo al archivo cfg.xml!!

Modificaciones Mapeo Users

User.java | *UserHistory.java | UserHistory.hbm.xml | User.hbm.xml | hibernate.cfg.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!-- Generated 20/11/2013 12:53:48 by Hibernate Tools 3.4.0.CR1 -->
<hibernate-mapping>
    <class name="programaSimple.User" table="USERS">
        <id name="id" type="int">
            <column name="ID" />
            <generator class="identity" />
        </id>
        <property name="name" type="java.lang.String">
            <column name="NAME" />
        </property>
        <property name="total" type="int">
            <column name="TOTAL" />
        </property>
        <property name="objetivo" type="int">
            <column name="OBJETIVO" />
        </property>

        <list name="History" table="USER_HISTORY" inverse="true" cascade="save-update">
            <key column="USER_ID" />
            <list-index column="POSITION" />
            <one-to-many class="programaSimple.UserHistory" />
        </list>
    </class>
</hibernate-mapping>
```

Nuevo Mapeo UserHistory

User.java | *UserHistory.java | UserHistory.hbm.xml | User.hbm.xml | hibernate.cfg.xml

```xml
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!-- Generated 20/11/2013 19:29:46 by Hibernate Tools 3.4.0.CR1 -->
<hibernate-mapping>
    <class name="programaSimple.UserHistory" table="USERHISTORY">
        <id name="id" type="int">
            <column name="ID" />
            <generator class="increment" />
        </id>
        <many-to-one name="user" class="programaSimple.User" not-null="true">
            <column name="USER_IDUSER" />
        </many-to-one>
        <property name="entryTime" type="java.util.Date">
            <column name="ENTRYTIME" />
        </property>
        <property name="detalle" type="java.lang.String">
            <column name="DETALLE" />
        </property>
    </class>
</hibernate-mapping>
```

No Olviden agregar este mapeo al archivo cfg.xml!!

Modificaciones Mapeo Users

User.java | *UserHistory.java | UserHistory.hbm.xml | User.hbm.xml | hibernate.cfg.xml

```xml
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!-- Generated 20/11/2013 12:53:48 by Hibernate Tools 3.4.0.CR1 -->
<hibernate-mapping>
    <class name="programaSimple.User" table="USERS">
        <id name="id" type="int">
            <column name="ID" />
            <generator class="identity" />
        </id>
        <property name="name" type="java.lang.String">
            <column name="NAME" />
        </property>
        <property name="total" type="int">
            <column name="TOTAL" />
        </property>
        <property name="objetivos" type="int">
            <column name="OBJETIVO" />
        </property>

        <list name="history" table="USER_HISTORY" inverse="true" cascade="save-update">
            <key column="USER_ID" />
            <list-index column="POSITION" />
            <one-to-many class="programaSimple.UserHistory" />
        </list>
    </class>
</hibernate-mapping>
```

Probemos el ejemplo!!

Mas Info?

http://www.hibernate.org/docs

ORM - Object Relational Mapping