#### hibernate.cfg.xml

Está compuesto de propiedades que le permiten realizar una conexión a una base de datos a través de JDBC. Estas propiedades tienen una cabecera principal de <hibernate-configuration>, donde se configura <session-factory>.

Este session-factory tiene las siguientes <property name=”?”>?</property>:

* **connection.driver\_class**: com.mysql.jdbc.Driver -> Driver utilizado para conectarse a la base de datos.
* **connection.url**: jdbc:mysql://127.0.0.1/bbdd -> Dirección de la base de datos. En versiones más antiguas a la 5.5.1 de Hibernate, es necesario incluir “?serverTimezone=UTC” para una conexión correcta.
* **connection.username**: usuario
* **connection.password**: contraseña
* **dialect**: org.hibernate.dialect.MySQL5Dialect -> Formato en el que se enviarán las sentencias SQL.
* **hibernate.default\_schema**: bbdd -> Base de datos predeterminada a la que se enganchará una vez realizada la conexión. Puede ser distinta a la base de datos a la que se conecta en un principio.
* **hibernate.show\_sql**: false/true -> Si queremos mostrar la sentencia SQL que termina ejecutando Hibernate.
* **hibernate.hbm2ddl.auto**: none/create/update -> Indica a Hibernate si queremos construir las tablas en la base de datos, actualizarlas o no cuando ejecutemos el programa.

#### Relaciones

* **1 a 1**
  + **XML**: <one-to-one name=”atributo” cascade=”all”/>
  + **Anotación**:

@OneToOne(cascade=CascadeType.ALL)

@PrimaryKeyJoinCOlumn

private Object atributo;

* **1 a X**
  + **Desordenada**
    - **XML**:
      * **Clase con 1 a X**

<set name=”atributoXa1” cascade=”all” inverse=”true”>

<key>

<column name=”nombreEnTabla1aX”/>

</key>

<one-to-may class=”claseXa1”/>

</set>

* + - * **Clase con X a 1**

<many-to-one name=”atributo1aX”>

<column name=”nombreTabla1aX”/>

</many-to-one>

* + - **Anotación**:
      * **Clase con 1 a X**

@OneToMany(mappedBy=”atributo1aX”, cascade=CascadeType.all)

private Set<ClaseXa1> atributoXa1;

* + - * **Clase con X a 1**

@ManyToOne

@JoinColumn(name=”nombreEnTabla1aX”)

private Clase1aX atributo1aX;

* + **Ordenada**
    - **XML**:
      * **Clase con 1 a X**

<list name=”atributoXa1” cascade=”all” inverse=”false”>

<key>

<column name=”nombreEnTabla1aX”/></key>

<list-index>

<column name=”idx”/>

</list-index>

<one-to-may class=”claseXa1”/>

</set>

* + - * **Clase con X a 1**

<many-to-one name=”atributo1aX”>

<column name=”nombreEnTabla1aX”/>

</many-to-one>

* + - **Anotación**:
      * **Clase con 1 a X**

@OneToMany(mappedBy=”atributo1aX”, cascade=CascadeType.all)

@JoinColumn(name=”nombreEnTabla1aX”)

@IndexColumn(name=”idx”)

private List<ClaseXa1> atributoXa1;

* + - * **Clase con X a 1**

@ManyToOne

@JoinColumn(name=”nombreEnTabla1aX”)

private Clase1aX atributo1aX;

* **X a X**
  + **XML**:
    - **Clase X1**

<set name=”atributoX2” table=”X1X2” cascade=”all” inverse=”true”>

<key><column name=”nombreEnTablaX1”/></key>

<many-to-many column=”nombreEnTablaX2” class=”claseX2”/>

</set>

* + - **Clase X2**

<set name=”atributoX1” table=”X1X2” cascade=”all” inverse=”true”>

<key><column name=”nombreEnTablaX2”/></key>

<many-to-many column=”nombreEnTablaX1” class=”claseX1”/>

</set>

* + **Anotación**:
    - **Clase X1**

@ManyToMany(cascade={CascadeType.ALL})

@JoinTable(name=”X1X2”, joinColumns={@JoinColumn(name=”nombreEnTablaX1”)}, inverseJoinColumns={@JoinColumn(name=”nombreEnTablaX2”)})

private Set<ClaseX2> atributoX2 = new HashSet();

* + - **Clase X2**

@ManyToMany(cascade={CascadeType.ALL}, mappedBy=”atributoX2”)

private Set<ClaseX1> atributoX1 = new HashSet();

#### Mapeo por anotaciones

#### Mapeo por XML

* **Id**:
* **Columna**:

#### Consultas HQL