
Exámen Hibernate+HQL

Tu nombre



Hibernate.cfg.xml

En el archivo de configuración de Hibernate ponemos:

- **El driver que vamos a utilizar** en este caso el **jdbc**.
- **Conexión a la BDD** con el nombre de la BDD "TipoA" (**previamente ejecutada**)
- **Ponemos nombre Usuario** de la BDD y la **contraseña** de la **BDD**
- **Indicamos el fichero donde hacemos el mapeo** que se llama **pokemon.gbm.xml**

```
1 <hibernate-configuration>
2   <session-factory>
3     <!-- Propiedades de conexión a la BBDD-->
4     <property name="connection.driver_class">com.mysql.jdbc.Driver</property>
5     <property name="connection.url">jdbc:mysql://localhost:3306/TipoA</property>
6     <property name="connection.username">root</property>
7     <property name="connection.password">Med@c</property>
8
9     <!-- Mapeo de entidades -->
10    <mapping resource="pokemon.hbm.xml"/>
11  </session-factory>
12 </hibernate-configuration>
13
```

Clase.hbm.xml

En el fichero del mapeo ponemos: Los datos de la clase Java (POJO)

- El nombre del fichero de la clase que es Pokemon
- Le ponemos una descripción
- **Ponemos los nombres de las variables para que sean mapeados.** (Como se llaman en la BDD y como queremos mapearlos) que luego pondremos cuando hagamos las consultas en Java. **Yo le he puesto el mismo**, para no confundirme y ponemos también de que tipo son los datos.

```
1 <hibernate-mapping>
2   <class name="com.example.demo.Pokemon" table="Pokemon">
3     <meta attribute="class-description">
4       Esta clase contiene información de los pokemon.
5     </meta>
6     <id name="idPokemon" type="int" column="idPokemon">
7       <generator class="native"/>
8     </id>
9     <property name="nombre" column="Nombre" type="string"/>
10    <property name="estatura" column="Estatura" type="int"/>
11    <property name="peso" column="Peso" type="int"/>
12  </class>
13 </hibernate-mapping>
14
```



Clase.java

En la clase (POJO):

Tenemos que crear:

- **Atributos** idPokemon, nombre, estatura y peso
- **Constructor vacío**
- **Métodos get y set de todos los atributos.**



```
1  public class Pokemon {
2
3      private Integer idPokemon;
4      private String nombre;
5      private Integer estatura;
6      private Integer peso;
7
8
9      public Pokemon() {
10     }
11
12     public Pokemon(String nombre, int estatura, int peso) {
13         this.nombre = nombre;
14         this.estatura = estatura;
15         this.peso = peso;
16     }
17
18     public Integer getIdPokemon() {
19         return idPokemon;
20     }
21
22     public void setIdPokemon(int idPokemon) {
23         this.idPokemon = idPokemon;
24     }
25
26     public String getNombre() {
27         return nombre;
28     }
29
30     public void setNombre(String nombre) {
31         this.nombre = nombre;
32     }
33
34     public Integer getEstatura() {
35         return estatura;
36     }
37
38     public void setEstatura(int estatura) {
39         this.estatura = estatura;
40     }
41
42     public Integer getPeso() {
43         return peso;
44     }
45
46     public void setPeso(int peso) {
47         this.peso = peso;
48     }
49
50
51
52 }
53
54
```



Resultados:

Todo funciona correctamente

Create Operation

Código / Ejecución

```
--- Menú CRUD Pokemon ---  
1. Crear Pokemon HQL  
2. Mostrar Pokemon HQL  
3. Actualizar Pokemon Nombre SQL  
4. Eliminar Pokemon SQL  
5. Salir  
Elija una opción: 1  
Crear nuevo Pokemon:  
Nombre: pokemon1  
Estatura: 3  
Peso: 45  
Pokemon creado con éxito.
```

Read Operation

Código / Ejecución

```
--- Menú CRUD Pokemon ---  
1. Crear Pokemon HQL  
2. Mostrar Pokemon HQL  
3. Actualizar Pokemon Nombre SQL  
4. Eliminar Pokemon SQL  
5. Salir  
Elija una opción: 2  
ID_POKEMON: 1, Nombre: Pikachu, Estatura: 20, Peso: 60  
ID_POKEMON: 2, Nombre: Snorlax, Estatura: 100, Peso: 200  
ID_POKEMON: 3, Nombre: Gyarados, Estatura: 500, Peso: 250  
ID_POKEMON: 4, Nombre: PokemonInventado, Estatura: 3, Peso: 45
```



Update Operation

Código / Ejecución

```
--- Menú CRUD Pokemon ---  
1. Crear Pokemon HQL  
2. Mostrar Pokemon HQL  
3. Actualizar Pokemon Nombre SQL  
4. Eliminar Pokemon SQL  
5. Salir  
Elija una opción: 3  
Inserte el código de pokemon:  
4  
Nuevo Nombre: PokemonInventado  
Pokemon actualizado con éxito.
```

Delete Operation

Código / Ejecución

```
--- Menú CRUD Pokemon ---  
1. Crear Pokemon HQL  
2. Mostrar Pokemon HQL  
3. Actualizar Pokemon Nombre SQL  
4. Eliminar Pokemon SQL  
5. Salir  
Elija una opción: 4  
Inserte el código de pokemon:  
4  
Pokemon actualizado con éxito.
```