

# P1-UT6 - Romper el hielo con acceso a BBDD

Se trata de continuar la práctica anterior realizando los apartados 6 y 7 del documento adjunto (P3 SpringQuizz.pdf del profesor Marcos Puente)

Se trata, como siempre, de documentar su realización entregando un documento en formato abierto donde describas los pasos realizados, problemas encontrados, soluciones aportadas y conclusiones sobre la consecución del objetivo concreto de cada apartado, además de un adjunto con el código fuente.

## Introducción

En esta practica he conectado la base de datos de libros que hicimos anteriormente con el Quiz realizado por mi mismo, y he conseguido adaptando los valores de la Base de datos con las variables del Quiz.

Entonces he conseguido que se cree un Quiz en el cual las respuestas además de guardarse en la BBDD se muestren en una página aparte.

## 6º Inserción de resultados en la base de datos:

Los resultados de cada jugador deben ser almacenado en la base de datos. De esta manera cada vez que un jugador acabe el quiz se le mostrará el resultado obtenido con su clasificación.

Los datos se guardan en una base de datos, que se crea por terminal manualmente. (la info esta en el archivo import.sql.)

Se añade por defecto un usuario como prueba de que la base de datos es funcional.

```
import.sql X
E: > Users > amado > Documents > workspace-spring-tool-suite-4-4.16.1.RELEASE > practicaEncuesta2 > src > main > resources > import.sql
1  -- INSERT INTO `users` (`id`, `nombre`, `casa`, `puntos`) VALUES (1, 'amado', 'Griffindor', '23');
2
3  -- Datos Manuales de la BBDD
4  DROP DATABASE IF EXISTS `casasHowarts`;
5  CREATE DATABASE `casasHowarts`;
6  DROP USER IF EXISTS `userMago` @`localhost`;
7  CREATE USER `userMago` @`localhost` IDENTIFIED BY 'user1234';
8  GRANT ALL PRIVILEGES ON *.* TO `userMago` @`localhost`;
9  USE `casasHowarts`;
10 CREATE TABLE `users` (
11     ID int PRIMARY KEY NOT NULL AUTO_INCREMENT,
12     NOMBRE VARCHAR (100) NOT NULL,
13     CASA VARCHAR (100) NOT NULL,
14     PUNTOS int NOT NULL
15 );
16 INSERT INTO `users` (`id`, `nombre`, `casa`, `puntos`) VALUES
17 ([1, 'amado', 'Griffindor', '23']);
18
```

Captura del código del uso de la BBDD

```
MariaDB [casasHowarts]> select * from users;
+-----+-----+-----+-----+
| ID | NOMBRE | CASA      | PUNTOS |
+-----+-----+-----+-----+
| 1 | amado  | Griffindor | 23      |
+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

Captura de la tabla con el insert añadido.

## 7º Cargar los resultados de otros usuarios:

Existirá una página a modo de tablón donde se mostrará las puntuaciones clasificación obtenida por los últimos 5 usuarios.

Para ello en el formulario hemos hecho unas consultas a los diferentes datos para que se puedan mostrar

```
:body>
<div class="listaUsuarios">
  <h2> Usuarios </h2>
  <table class="table">
    <thead>
      <tr>
        <th scope="col">Id</th>
        <th scope="col">Usuario</th>
        <th scope="col">Casa</th>
        <th scope="col">Puntos</th>
        <th scope="col">Borrar</th>
        <th scope="col">Modificar</th>
      </tr>
    </thead>
    <tbody>
      <tr th:each="user: ${users}">
        <td th:text="${user.id}" />
        <td th:text="${user.nombre}" />
        <td th:text="${user.casa}" />
        <td th:text="${user.puntos}" />
        <td><a class="btn btn-danger borrar" th:href="@{/borraruser/{id}(id=${user.id})}"><i
          class="fas fa-trash"></i></a></td>
        <td><a class="btn btn-success editar" th:href="@{/editaruser/{id}(id=${user.id})}"><i
          class="fas fa-edit"></i></a></td>
      </tr>
    </tbody>
  </table>
  <form th:action="@{/destroy}" method="post">
    <input type="submit" value="Volver al Inicio" />
  </form>
</div>
:/body>
```

Captura del código del tablón



Id	Usuario	Casa	Puntos	Borrar	Modificar
1	amado	Griffindor	23		

Volver al Inicio

Captura del tablón

## **Añadidos para subir nota:**

### **Pruebas ¿qué pasa si no existe la tabla?**

Sobre la tabla, hice la prueba de quitar la tabla de la base de datos y ejecuté el programa dio y siguiente error:

```
java.sql.SQLException: Table 'casashowarts.users' doesn't exist in engine
at com.mysql.cj.jdbc.exceptions.SQLException.createSQLException(SQLException.java:120) ~[mysql-connec
at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.
at com.mysql.cj.jdbc.ClientPreparedStatement.executeInternal(ClientPreparedStatement.java:916
at com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdateInternal(ClientPreparedStatement.java
```

Captura del error al no haber tablas.

Ocurre porque el proyecto no tiene una funcionalidad para generar tablas con los datos introducidos, ya que no era del estilo que lo había enfocado, pero en otros proyectos mediante darle simplemente el insert de la tabla, se crea sin tener que haber ninguna en la Base de datos.

## **Implementar borrado y/o modificación de clasificaciones.**

### **Implementar borrado de usuarios**

Sobre el Borrado de los usuarios he creado una función que los identifica por id y los borra totalmente de la BBDD mediante el clic de un botón.

```
@RequestMapping(value = "/borraruser/{id}", method = RequestMethod.GET)
public String borraruser(@PathVariable("id") long idUser) {

    userDao.deleteById(idUser);

    return "redirect:/clasificacion";
}
```

Captura de la función de borrado



Id	Usuario	Casa	Puntos	Borrar	Modificar
1	amado	Griffindor	20		

Captura de la versión web del icono de borrado

## Implementar Modificación de los usuarios

Sobre la modificación de usuarios, he creado dos funciones las cuales se encargan de ese proceso, el cual envía al usuario a una pagina diferente para rellenar los nuevos datos y al modificarlo, se puede observar el cambio en la BBDD y en la web misma.

```
@RequestMapping(value = "/editaruser/{id}", method = RequestMethod.GET)
public String editaruser(@PathVariable("id") long idUser, Model modelo) {

    Optional<User> user = userDao.findById(idUser);

    if (user == null) {
        return "clasificacion";
    }
    User u1 = user.get();

    modelo.addAttribute("user", u1);
    return "editaruser";
}

@RequestMapping(value = "/editaruser/{id}", method = RequestMethod.POST)
public String modificaruser(@PathVariable("id") long idUser, @RequestParam String nombre, @RequestParam String casa,
    @RequestParam Integer puntos) {

    User user = new User(idUser, nombre, casa, puntos);
    userDao.update(user);

    return "redirect:/clasificacion";
}
```

Captura de las funciones de modificar los usuarios



Captura del usuario sin modificar

```
MariaDB [casasHowarts]> select * from users;
+-----+-----+-----+-----+
| ID | NOMBRE | CASA      | PUNTOS |
+-----+-----+-----+-----+
| 1 | amado  | Griffindor | 23      |
+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

Captura de la BBDD sin modificar

## Modificar Usuario

ID:

1

Nombre:

amado

Casa:

Griffindor

Puntos:

20

Modificar Usuario

Página de modificación de usuarios



Id	Usuario	Casa	Puntos	Borrar	Modificar
1	amado	Griffindor	20		

Captura del usuario modificado

```
MariaDB [casasHowarts]> select * from users;
+-----+-----+-----+-----+
| ID | NOMBRE | CASA       | PUNTOS |
+-----+-----+-----+-----+
| 1  | amado  | Griffindor | 20     |
+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

Captura de la BBDD del usuario modificado

## Monitorización de consultas

Usar el monitor de qué consultas se hacen en tiempo real a la base de datos: <https://mkyong.com/spring-boot/spring-boot-show-hibernate-sql-query/>

Sobre la monitorización de las consultas, he intentado hacer la consulta, pero no ha dado las señales que necesitaba, se cree que puede ser un problema de versiones o alomejor de algo mas profundo, al no dar ninguna señal de error ni nada la verdad es difícil saber cual es el problema. Le adjunto la captura de las líneas colocadas como en el tutorial esta descrito, pero no dan ningún resultado.

```
1 spring.datasource.url=jdbc:mysql://localhost/casasHowarts?  
2 spring.datasource.username=userMago  
3 spring.datasource.password=user1234  
4 spring.sql.init.mode=always  
5  
6 #show sql statement  
7 logging.level.org.hibernate.SQL=debug  
8  
9 #show sql values  
10 logging.level.org.hibernate.type.descriptor.sql=trace
```

Captura del código del tutorial usado en el proyecto

## Conclusión

La practica ha sido muy completa, creo que ha faltado tiempo para explicar algunas formas o dar como una pequeña idea de como atajar puntos como los jars o otras Bases de datos desde mi punto de vista.