

Documentación del proyecto (Siete y medio)

Álvaro Terrasa Moya – 1º GS DAM

1. Cómo se ha realizado

El proyecto comprende varias herramientas que se han usado para su realización:

1. Maven: Permite facilitar muchas partes de la construcción de un proyecto, como las dependencias, ciclos de vida, arquetipos, versiones de las librerías...

Todo ello se controla a partir de un directorio del usuario, donde se almacenan y gestionan todos los datos que se usan para el proyecto en cuanto a las características anteriores. A su vez, el archivo 'pom.xml' permite gestionar todo esto en la interfaz del IDE.

2. IntelliJ: Es un IDE que facilita mucho la programación en Java, y ofrece características que mejoran mucho la calidad de vida en la programación.

3. Tomcat 10: Permite realizar pruebas en servidor local de los proyectos de forma muy simple: tan sólo descargando un fichero de Tomcat de su página y realizando la configuración de ejecución.

4. JSP: Renderiza las páginas en el servidor y las muestra directamente en el cliente. Es una manera efectiva de juntar HTML, CSS y JavaScript. Permite inyectar código de Java directamente, pero no es una práctica actual, por lo que no lo he hecho.

5. Servlets: Son clases de Java que implementan HTTP, y permiten hostear aplicaciones web en Java. Permiten controlar el flujo de peticiones y respuestas, y controlar muchos aspectos del aplicativo, como la sesión, cookies, API, filtros...

6. JAX-RS REST API: Un conjunto de clases, aisladas en un proyecto aparte del proyecto web, que permite tener una arquitectura RESTful en el proyecto. Los controladores manejan las peticiones a la API, el servicio controla el manejo de las conexiones a la base de datos y el flujo entre el controlador y el manager; y el manager se encarga de realizar las operaciones con la base de datos. Esto permite dividir de forma muy cómoda todos los aspectos que intervienen en una API REST.

A su vez, aquí se encuentra todo lo pertinente a la realización de la conexión con la base de datos (connector y propiedades).

7. DAO y DTO: Son objetos que representan el acceso a la base de datos (DAO) y la transferencia desde la base de datos (DTO).

8. web.xml: Es un archivo que permite añadir información que se usará para el funcionamiento del aplicativo web, como datos, mapeos de filtros y servlets...

2. Lenguajes de programación usados

Los lenguajes de programación de los que he hecho uso son:

1. Java: Lógica del servidor web (servlets, servicios y cliente) y API (controlador, servicio y manager).
2. HTML, CSS y JavaScript: Apariencia y lógica de la implementación del programa.
3. SQL: Gestionar la base de datos.
4. XML: Configuración y manejo de los archivos .xml.

3. Base de datos usada

Para la base de datos he usado MySQL Workbench, que me permite crear esquemas y tablas para su uso en el programa. Prácticamente toda la configuración de la base de datos la he realizado a través de GUI.

En concreto, he creado una tabla de jugadores (player) y otra de partidas (game). Estas son gestionadas en el proyecto de REST API.

4. Información sobre el funcionamiento del programa

Básicamente, para poder desplegar el programa, se descarga Maven y Tomcat, y se configuran en IntelliJ. En Maven añades las dependencias y todo lo necesario para el proyecto, y en Tomcat configuras el servidor. A partir de ahí, se puede empezar a programar por donde uno quiera; lo importante es ir conectándolo todo.

De forma general, el “camino” que sigue el programa es:

1. Servlets: Se reciben las peticiones y se mandan las respuestas. Lo que haya entre esos dos puntos depende del servlet, por lo que se podría realizar cualquier tipo de operación en medio para asegurar que se responde lo que se desea al cliente.
2. Servicios web: Obtienen peticiones de los servlets, y las redirigen a los clientes para realizar operaciones de API.
3. Clientes web: Obtienen las URL de la API, y se aseguran de que desde el servidor web se manda y se recibe todo de forma correcta.
4. Controlador: Se conecta con el cliente web para gestionar lo que entra y sale de la API (similar al anterior). Además, llama a los servicios JAX-RS para realizar las operaciones, mandando lo pertinente para ello, y define las URL de la API.

5. Servicios JAX-RS: Se encargan de gestionar la conexión con la BBDD, llamar a las operaciones de esta y mandar la respuesta al controlador.

6. Manager: Implementa la funcionalidad de gestión de la BBDD, en cuanto a operaciones SQL.