Bachelor's degree in Computer Science and Engineering
Tecnologías Informáticas para la Web
2023-2024

Práctica 3

"Plantilla de una liga de fútbol"

Luis Daniel Casais Mezquida - 100429021
Pablo Montero Poyatos - 100429163
Javier Moreno Yébenes - 100428998
Alberto Urbano Ballesteros - 100429046

Professor:
Jesús Hernando Corrochano

Tabla de contenidos

texlua: security risk: running with elevated privileges

1 Introducción

La consiguiente práctica consistirá en la última versión de la aplicación desarrollada en la anterior entrega, la cual utilizaba una base de datos relacional, desarrollada en MySQL y con el uso de JPA o JDBC entre otras herramientas. En este caso, se utilizarán tecnologías como las bases de datos Not Only SQL (NoSQL) utilizando MongoDB y se cambiará la estructura interna de la aplicación para tener un producto más orientado el mercado actual.

La aplicación web realizada está basada en la posibilidad de agregar, editar y eliminar jugadores que podrían formar parte de la plantillas de los clubes más prestigiosos de España (eg: Atlético de Madrid, *er* Beti, Rayo Vallecano, F.C. Barcelona... etc). Además de esto, se ha añadido la posibilidad de seleccionar un equipo favorito, la creación de un inicio de sesión (con cuentas con privilegios de administrador y de usuario) y un pequeño sistema de chat con el objetivo de permitir a los usuarios interactuar entre sí.

El avance de esta aplicación respecto a la versión anterior es la implementación de los llamados "microservicios", que nos permiten tener una aplicación más fácil de mantener, mejor encapsulada y más resistente a errores gracias a esta independencia entre funcionalidades.En estas estructuras, hay un módulo principal que utiliza las distintas funcionalidades que le proporcionan los demás servicios, de forma que este pueda desempeñar correctamente las tareas que se le requieren a la aplicación.

En nuestro caso, este mödulo principal es el "front" (frontend) es el que utilizará los distintos Web Services que ofrecen los otros módulos. Del módulo equiposWS utiliza las operaciones de gestión del banco de jugadores, del de loginWS todo aquello relacionado con el inicio de sesión y la gestión de usuarios y por último, a chatWs le solicita todas las funcionalidades que respectan al servicio de mensajería.

A rasgos generales, ya que en el proyecto aparecen numerosos archivos de configuración y propiedades (pom.xml, los .properties, los .jar, mvnw ... etc) esta es la estructura general del proyecto, y estos son sus archivos más relevantes:

2 Front

Se trata de el frontend de la aplicación, y aquel y hace uso de todos los Web Services que tiene nuestra aplicación, para funcionalidades como el login, los mensajes o mostrar jugadores o equipos, entre otras muchas cosas.

2.1 "java/es/uc3m/tiw"

Carpeta que posee todos los archivos de java que utiliza frontend (sin contar los microservicios). Hay principalmente un controlador y la definición de entidades necesarias para el

programa. Esta carpeta es en cierta forma el backend (junto a otras funcionalidades), que es principalmente lo que el usuario no ve pero la aplicación ejecuta.

2.1.1 Controllers

Esta carpeta solo contiene el controlador del programa en general "MainController.java". Este es al archivo más importante del programa ya que es en el cual se utilizan todos los microservicios creados por los módulos de la aplicación, es el encargado de llamar a todos ellos y utilizarlos correctamente. Con ayuda de los Mappings es capaz de redirigir a todas las páginas que tiene nuestra aplicación (html y css).

2.1.2 Entities

La definición de todas las entidades que pueden aparecer en el frontend y en la gestion de la base de datos de los jugadores. Aqui se definen los siguientes objetos, con un archivo java para cada uno. Los objetos que representan a los administradores y a los usuarios, y sus respectivas contraseñas, los jugadores, las posiciones y las plantillas. Todos estos objetos se definen por separado y explican la información que se quiere guardar de cada uno por ejemplo en el usuario su correo, su nombre, sus apellidos ... etc, o en un jugador su dni, su posición, equipo...

2.1.3 "frontApplication.java"

Archivo encargado de definir la inicialización de la aplicación del frontend.

2.2 Resources

Carpeta que contiene la parte del HTML y CSS de nuestra aplicación, son principalmente la parte visual de nuestra herramienta, con la cual el usuario interactúa, y que conectada con las herramientas de la carpeta anterior hacen funcionar la aplicación. Esta carpeta es que lo que se puede considerar como el frontend.

Esta capeta se divide en templates, en la que están los HTML mas grandes que representan páginas enteras, styles, que ya contiene los archivos css y la acarpeta components que tiene tanto los HTML como lo css de componentes pequeños de la página como pueden ser algún tipo de botones.

3 EquiposWS

Carpeta que contiene todo lo relacionado con el microservicio de la gestión de la base de datos de los jugadores, a la que tienen acceso todos los usuarios con distintos permisos.

3.1 db_create.sql

Cada microservicio tiene una base de datos propia, este archivo se encarga de inicializar la de este microservicio.

3.2 Controllers

Solo contiene el archivo MainController.java encargado de administrar todas las funcionalidades de este servicio. Este archivo utiliza los mappings para redireccionar correctamente entre las páginas y realizar las pertinentes llamadas.

3.2.1 Entities

La definición de todas las entidades que pueden aparecer en la base de datos de los jugadores. Aqui se definen los siguientes objetos, con un archivo java para cada uno. Los objetos que representan los jugadores, las posiciones,los equipos,las plantillas y las keys de cada plantilla. Todos estos objetos se definen por separado y explican la información que se quiere guardar de cada uno por ejemplo en el usuario su correo, su nombre, sus apellidos ...etc

3.2.2 Repositories

Definición de los DAO(Data Access Object) necesarios para el acceso a la base de datos de este microservicio. Se tiene EquipoDAO.java, JugadorDAO.java, PlatillaDAO.java y PosicionDAO.java representando sus correspondientes entidades.

3.2.3 Utils

Carpeta con herramientas que es útil tener aisladas en funciones u objetos para simplificar el código. En este caso solo se tiene un archivo que valida si un DNI es correcto.

4 LoginWS

Carpeta que contiene todo lo relacionado con el microservicio del login, los administradores y los usuarios, y sus distintos permisos.

4.1 db_create.sql

Cada microservicio tiene una base de datos propia, este archivo se encarga de inicializar la de este microservicio.

4.2 Controllers

Solo contiene el archivo MainController.java encargado de administrar todas las funcionalidades de este servicio. Este archivo utiliza los mappings para redireccionar correctamente entre las páginas y realizar las pertinentes llamadas.

4.2.1 Entities

La definición de todas las entidades que pueden aparecer en el login. Aqui se definen los siguientes objetos, con un archivo java para cada uno. Los objetos que representan los administradores, a los usuarios y a sus respectivas contraseñas. Todos estos objetos se definen por separado y explican la información que se quiere guardar de cada uno por ejemplo en un jugador su dni, su posición, equipo...

4.2.2 Repositories

Definición de los DAO(Data Access Object) necesarios para el acceso a la base de datos de este microservicio. Se tiene AministradorDAO.java, AdinPasswordDAO.java, UserDAO.java y UsuarioPassworsdDAO.java representando sus correspondientes entidades.

4.2.3 Utils

Carpeta con herramientas que es útil tener aisladas en funciones u objetos para simplificar el código. En este caso solo se tiene un archivo que valida si un DNI es correcto.

5 ChatWS

Carpeta que contiene todo lo relacionado con el microservicio del chat, utiliza los correos de ambas partes y almacena el mensaje.

5.1 db_create.sql

Cada microservicio tiene una base de datos propia, este archivo se encarga de inicializar la de este microservicio.

5.2 Controllers

Solo contiene el archivo MainController.java encargado de administrar todas las funcionalidades de este servicio. Este archivo utiliza los mappings para redireccionar correctamente entre las páginas y realizar las pertinentes llamadas.

5.2.1 Domains

La definición de todas las entidades que pueden aparecer en el chat.En este caso, la única entidad necesaria de almacenár más de un atributo es mensajes, con la persona que lo envía el cuerpo del mensaje y el cuerpo de este.

5.2.2 Repositories

MensajeRepository.java es el encargado de representar la base de datos en este caso. Es el encargado de crear las queries y funciones necesarias para realizar las distintas funcionalidades

6 Conclusiones

La realización de esta práctica ha supuesto un reto para el equipo y en ciertas ocasiones una ocasión perfecta para aprender a enfrentarnos a problemas reales que pueden aparecer en el desarrollo de una aplicación real de misma naturaleza. Creemos que el hecho de incluir una base de datos real del tipo NoSQL nos prepara para la forma de trabajar del mercado laboral, ya que son tecnologías que están en el día a día de aplicaciones de este estilo.

Además, ha sido un gran descubrimiento el aprender de la existencia de los microservicios, que de una forma muy simple (divide y venderás) es capaz de mejorar el mantenimiento, la encapsulación y la gestión de errores entre otros muchos aspectos, de forma verdaderamente significativa.