SDI – Sistemas Distribuidos e Internet

ENUNCIADO PRÁCTICA 1 – SPRING

INFORME

Grupo 2021-1005

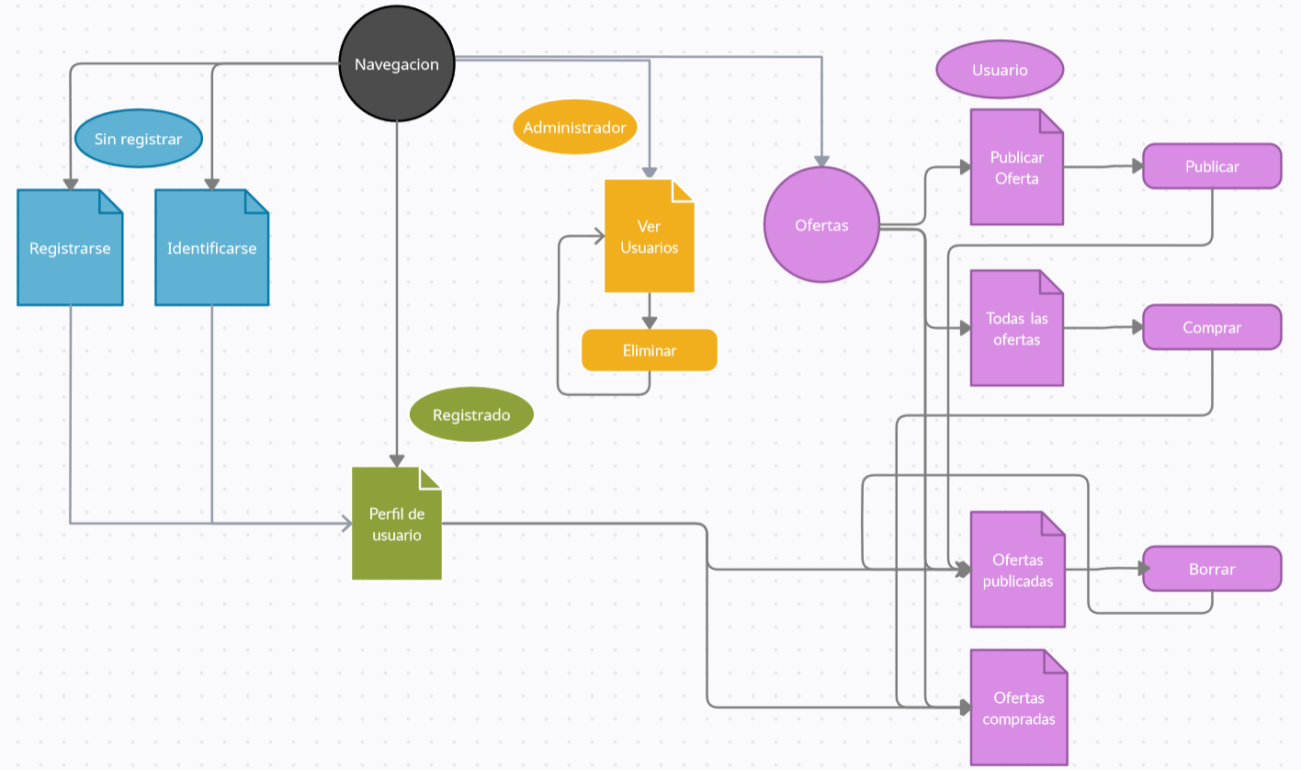
Nombre:	Samuel
Apellidos:	Moreno Vincent
Email:	UO266321@uniovi.es
Cód. ID GIT	2021-1005
% Participación	100%
Repositorio GitHub	https://github.com/samuelmorenov/sdi-entrega1-2021-1005

Índice

Introducción

Para la realización de este proyecto se ha utilizado como base el proyecto de Spring Boot realizado en el desarrollo de las clases prácticas de la asignatura, siguiendo los guiones de las sesiones de la 2 a la 5. Así pues, se ha mantenido toda la apariencia visual que tenía dichas prácticas. Además, se ha utilizado parte del proyecto de My Social Network entregado en la convocatoria adelantada en enero, utilizándolo únicamente como guía en caso de alguna duda para una mayor rapidez a la hora de realizarlo.

Mapa de navegación



Leyenda:

Pagina: Muestra cada una de las paginas existentes en la aplicación.

Permisos: Muestra los permisos requeridos para esa zona de la aplicación (código de colores explicados a continuación)

Acción: Posibles acciones a realizar en cada página.

Menú de navegación: Menú o submenú de navegación que se encuentra en la parte superior de la aplicación.

Flecha rellena: Indica un cambio de pagina accionado por el usuario.

Flecha vacía: Indica una redirección automática de la aplicación.

Código de colores:
Negro: Menú de navegación superior visible desde cualquier parte de la aplicación.
Azul: Opciones publicas para el registro e inicio de sesión del usuario.
Verde: Página común a todos los usuarios autentificados en la aplicación.
Amarillo: Página y acciones exclusivas del administrador.
Morado: Página y acciones exclusivas de usuarios.

Explicación del mapa

Al acceder a la aplicación el usuario no estará autentificado en la aplicación, por lo que desde el menú "Navegación" solo podrá acceder a las paginas de registro e identificación. Por defecto la aplicación le redirigirá a la pantalla de identificación, pero si no tiene un registro previo puede acceder a la pantalla de registro desde el menú. Una vez autentificado (ya sea como registrado o identificado) la aplicación redirige automáticamente a la pagina de perfil de usuario, que es común para todos los tipos de usuario.

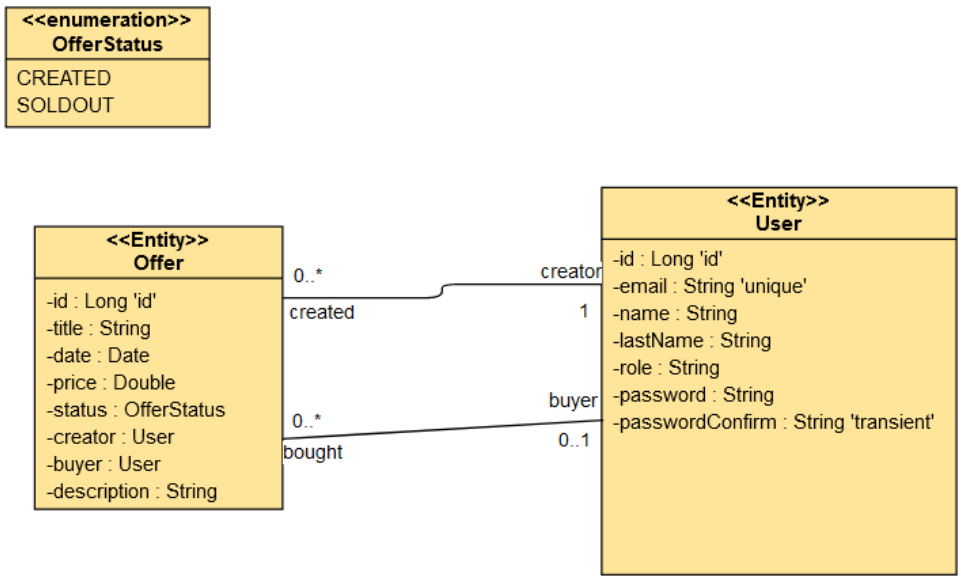
Si el usuario es administrador en la pantalla de perfil de usuario únicamente verá un mensaje de bienvenida, y en la barra de navegación tendrá la opción de ir a la página de usuarios registrados. En esta pagina el administrado puede marcar los usuarios de la lista que quiera y eliminarlos del sistema, cuando haga esto verá como los usuarios desaparecen de la lista.

Si el usuario es un usuario estándar en la pantalla de perfil de usuario le aparecerá su nombre, la cantidad de dinero que tiene, y dos enlaces para ver sus ofertas publicadas y compradas. Además, desde el menú de navegación podrá acceder a cuatro opciones:

- Publicar oferta: En esta pantalla el usuario podrá rellenar el formulario para la creación de una nueva oferta. Una vez relleno, al dar a publicar, se le redirigirá a la pantalla de ofertas publicadas automáticamente.
- Todas las ofertas: En esta pantalla por defecto se realiza una búsqueda de todas las ofertas existentes en el sistema, pudiendo filtrarlas con la barra de búsquedas. En el listado de las ofertas se muestra el título, la descripción, la fecha de publicación y las opciones de compra. Estas opciones estarán bloqueadas si la oferta ya ha sido comprada o si la oferta esta publicada por el usuario que esta navegando, y estará habilitada la opción de compra en caso contrario. Si el usuario compra la oferta se redirigirá automáticamente a la pantalla de ofertas compradas.
- Ofertas publicadas: En esta pantalla el usuario podrá ver todas las ofertas que ha publicado, y podrá borrar cualquiera de ellas. Cuando borre alguna podrá ver la lista de nuevo en la que no aparecerá esa oferta.
- Ofertas compradas: En esta pantalla el usuario podrá ver una lista de todas las ofertas que ha comprado.

Aspectos técnicos y de diseño relevantes

Modelo de dominio



El modelo de dominio es bastante sencillo, unicamente consta de ofertas (Offer) y usuarios (User).

Las ofertas tienen los atributos id (identificador), title (titulo de la oferta), description (breve descripcion de la oferta), date (fecha de alta de la oferta), price (precio en euros de la oferta), status (enumerado que dice si esta comprada o no), creator (usuario que publica la oferta) y buyer (usuario que compra la oferta).

Los usuarios tienen los atributos id (identificador), email (correo electrónico, que no puede estar repetido), name (nombre del usuario), lastName (apellidos del usuario), role (si es administrador o usuario estándar), password (contraseña del usuario que se guardara cifrada) y paswordComfirm (que no se guarda en la base de datos, pero es necesario para la validación de la contraseña).

Estos se relacionan de manera que cada usuario puede tener varias ofertas publicadas y compradas, pero cada oferta es publicada por un usuario y comprada por otro usuario.

Decisiones de implementación

Toda la arquitectura y diseño de la aplicación está basada en la aplicación realizada durante las practicas. En el único sitio donde hay cambios significativos es en el diseño de las pruebas. En lugar de estar todos apilados en una única clase se han separado en 13 clases distintas correspondientes a cada uno de los ejercicios. Todos ellos se ejecutan de forma consecutiva desde la clase “SdiEntrega120211005ApplicationTests.java”.

Esto conlleva el inconveniente de no poder ejecutar una prueba de forma individual a priori, ya que no ejecutaría el método “BeforeClass”. Para solucionar este problema, y además hacer la programación de las pruebas más fluida se ha implementado un patrón Singleton para el driver

"WebDriver" de selenium, de manera que ahora todos los test llaman a la misma instancia del driver y se inicializa una única vez. Este patrón se puede ver en la clase DriverSingleton.java.

Además, todas las clases de pruebas heredan de una clase BaseTests que establece el WebDriver, navega hasta la url de inicio antes de cada prueba y limpia las cookies al finalizarlas.

Información necesaria para el despliegue y ejecución

Antes de nada, hay que arrancar la base de datos. Esta está en la carpeta hsqldb, para arrancarla únicamente ejecutamos el archivo run.bat.

Es importante destacar que al arrancar la aplicación de Spring Boot la base de datos se reinicia. Esto se hace para que se puedan ejecutar las pruebas. Si se desea mantener la base de datos entre distintas veces hay que comentar o borrar la clase InsertSampleDataService.java y en el archivo application.properties cambiar la propiedad spring.jpa.hibernate.ddl-auto a validate.

Por lo demás únicamente hay que ejecutar la clase MySocialNetwork.java como Spring Boot App.

Para ejecutar las pruebas debe estar desplegada la aplicación y la base de datos. Se ejecutará la clase MySocialNetwork_Tests.java como JUnit Test

Conclusión

Spring Boot me parece una herramienta muy útil ya se utiliza mucho en el ámbito laboral por las empresas, pero siento que falta la distinción entre frontend y backend, el uso de las plantillas Thymeleaf complica mucho la comunicación entre estas partes, habría preferido aprender a utilizar frameworks como Angular o React.