



SDI – Sistemas Distribuidos e Internet

ENUNCIADO PRÁCTICA 1 – SPRING

INFORME

Grupo 604-607

Nombre1:	Thalía
Apellidos1:	Cuetos Fernández
Email1:	UO264545@uniovi.es
Cód. ID GIT	1920-604
%Participación	50%
Nombre2:	Sonia
Apellidos2:	García Lavandera
Email2:	UO263536@uniovi.es
Cód. ID GIT	1920-607
%Participación	50%



Índice

INTRODUCCIÓN	3
MAPA DE NAVEGACIÓN	3
ASPECTOS TÉCNICOS Y DE DISEÑO RELEVANTES.....	4
INFORMACIÓN NECESARIA PARA EL DESPLIEGUE Y EJECUCIÓN	5
CONCLUSIÓN	5

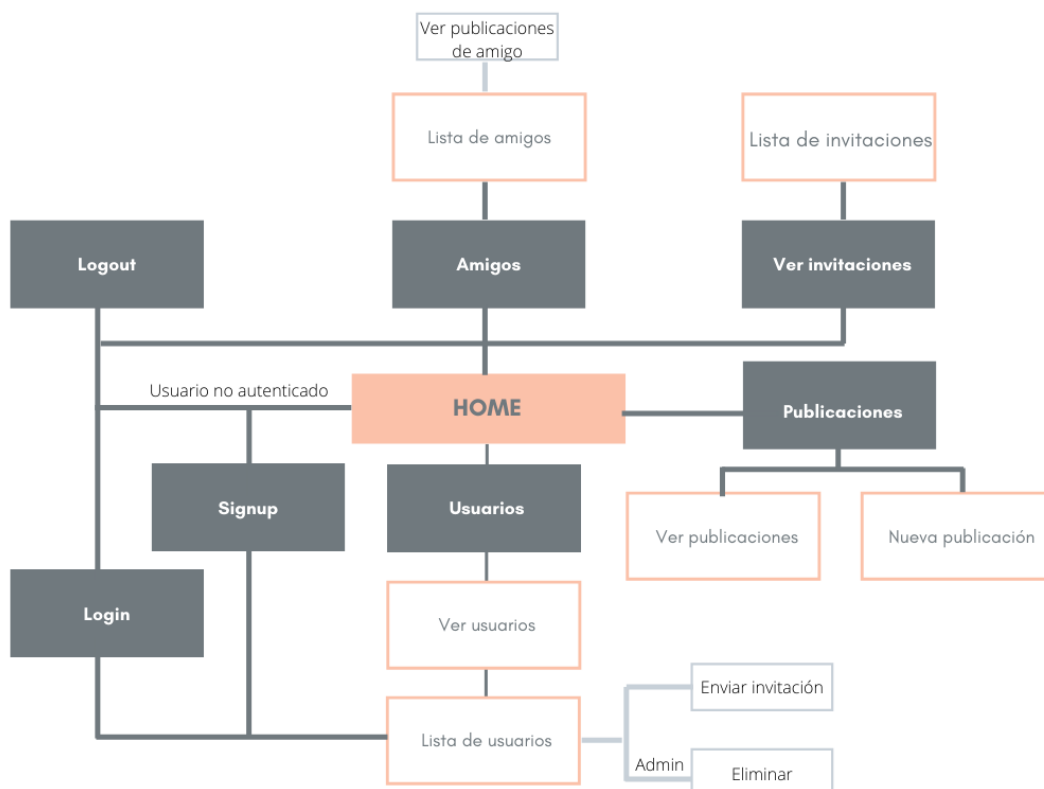


Introducción

Nuestra aplicación se llama Gold SocialNetwork y permite que tanto usuarios como administradores puedan hacer uso de la aplicación. En este caso, tanto los usuarios estándar como los administradores pueden hacer uso completo de la aplicación, es decir, los usuarios administradores podrán tener todas las funcionalidades al igual que los usuarios estándar: listar amigos, ver publicaciones del usuario en sesión y de sus amigos, ver invitaciones de amistad... Los usuarios pueden enviar invitaciones de amistad a otros usuarios y cuando estos acepten la invitación serán amigos. Además de esto, los usuarios pueden listar a los demás usuarios de la aplicación, aunque esto puede cambiar en función de si el usuario es estándar o si es administrador se verán listados distintos. En cuanto a la seguridad de la aplicación los usuarios estándar no podrán ver las opciones del administrador de forma que no podrán acceder a las urls de tipo /admin/*. Nosotras decidimos hacer un logo por iniciativa propia y ponerle un nombre a nuestro proyecto para que se vea nuestra implicación en ella y el trabajo que le hemos dedicado.

Nota: Tenemos que aclarar que realizamos tanto los requisitos obligatorios como los opcionales, así como las pruebas correspondientes.

Mapa de navegación





Aspectos técnicos y de diseño relevantes

En cuanto a aspectos técnicos nuestra aplicación está desarrollada en spring boot con una base de datos hsqldb donde se guardan los datos de la aplicación. Las pruebas se realizan con el entorno de desarrollo de Selenium usando pruebas JUnit 4. Con respecto a los patrones arquitectónicos se emplean el patrón MVC combinado con el N-layers. Además, para las vistas de la aplicación se usa el motor de plantillas Thymeleaf visto en clase.

A partir de aquí voy a desarrollar más detenidamente algunos aspectos técnicos del desarrollo.

Por una parte, estaría el desarrollo de las **entidades**. Aquí tenemos que destacar que la entidad User dispone de una lista de usuarios que son los amigos. Esto, aunque parece muy lógico como lo pensamos se nos complicó un poco cuando teníamos que eliminar múltiples usuarios ya que como tenían relaciones de amigos unos con otros, entraba como en una especie de bucle y si probabas a eliminar un usuario te eliminaba todos los usuarios no solo esto. Para solucionarlo en el servicio de users realicé un método que me eliminaba primero en el usuario de esos amigos del usuario en sesión y después eliminaba el usuario. A parte de eso en las pruebas me di cuenta de que tenía que usar un iterador ya que saltaba una excepción concurrentmodificationexception ya que estaba modificando una lista sobre la que estaba iterando.

Por otra parte, en los **controladores** en algún caso tuvimos que comprobar si los usuarios son amigos por ejemplo en el caso de listar las publicaciones de un amigo ya que tendremos que dar un error de autorización si intentas acceder a las publicaciones de un usuario que no es amigo.

En el caso de los **servicios** como en los **repositorios**, no tuvimos grandes problemas salvo el caso de eliminar usuarios como ya expliqué anteriormente.

Cuando el usuario se autentica correctamente la web le redirige a una página u otra en función de su rol: administrador o usuario. Para ello hemos tenido que crear la clase CustomAuthenticationSuccessHandler que implementa la interfaz AuthenticationSuccessHandler, añadirla como Bean e indicar que es el handler de la función login. Esta clase sobrescribe el método onAuthenticationSuccess, que es donde se comprueba qué rol tiene el usuario que se acaba de autenticar.

En cuanto al diseño:

Tuvimos que realizar una vista llamada **error.html** con un h2 y un h3 donde se informa el error de acceso de autorización por ejemplo que se usa cuando accedes a las publicaciones de un usuario que no es tu amigo.

Además, tenemos **dos vistas distintas para el listado de los usuarios** por parte del administrador y otra por parte de los usuarios estándar ya que los usuarios estándar no pueden eliminar usuarios mientras que los administradores sí que pueden. En el caso de los usuarios estándar se visualizará si son amigos o si se puede enviar una solicitud de amistad, pero en el caso de los administradores solo se puede eliminar usuarios seleccionándolos previamente mediante checkbox y luego pulsando el botón eliminar.



Información necesaria para el despliegue y ejecución

En primer lugar, tenemos que recordar que tenemos que tener instalado Java 8 ya que vamos a tener problemas al querer ejecutar las pruebas.

A continuación, lo que tenemos que hacer es arrancar la base de datos desde runServer.bat que se encuentra en la carpeta bin de hsqldb de esta forma nuestra base de datos estaría arrancada. A continuación, debemos arrancar la aplicación como Run as -> Java Application y a continuación si queremos ejecutar las pruebas hacemos Run as -> JUnit Test. Un aspecto importante a tener en cuenta es que no podemos ejecutar las pruebas sin tener arrancada la aplicación y no podemos arrancar la aplicación sin tener la base de datos funcionando.

A la hora de ejecutar los test se requiere que los test sean JUnit 4 ya que por defecto se configura con JUnit 5.

Si quisiéramos probar la aplicación accediendo a la url <http://localhost:8090/>

Nota importante: ya que no sabíamos cómo poner la ruta en la prueba 29 pues dejamos una ruta del tipo “C:\\...” por eso esa prueba va a fallar ya que el equipo donde se ejecuten esos tests no disponen de esa imagen.

Conclusión

Como conclusión podríamos decir que la aplicación nos quedó bastante bien y que funciona correctamente. Además, trabajamos horas en ella y realizamos tanto los ejercicios obligatorios como los opcionales. , implementamos todas las pruebas pedidas y funcionan, intentamos hacerlas lo más exhaustivas posibles, esperemos haberlo conseguido. En general, trabajamos bien como equipo y nos dividimos el trabajo más o menos con el mismo.