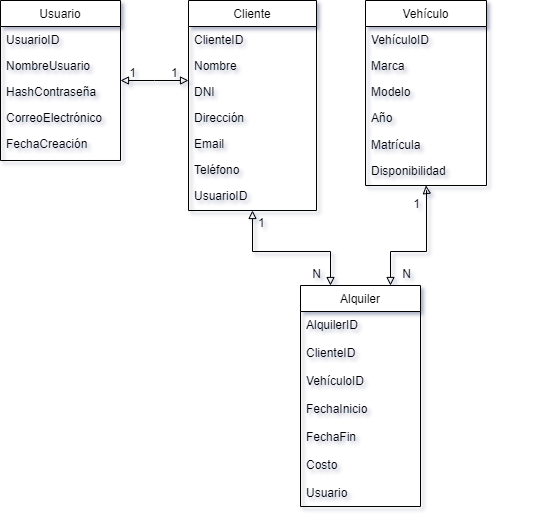
L

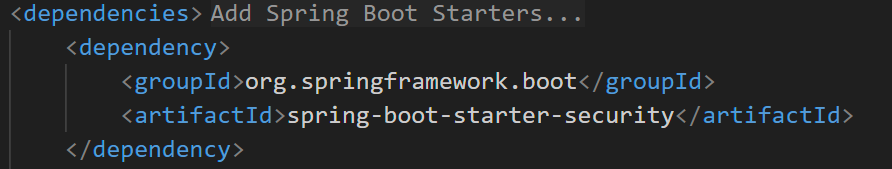
|  |  |
| --- | --- |
| Producto 3. Seguridad y usuarios  FP.065 Aplicación back-end con Java en servidores de aplicaciones | |
|  | |
|  | Índice  1 - Modificar nuestro diagrama E/R para la incorporación de usuarios a nuestra aplicación tanto desde el punto de vista de la seguridad como desde el punto de vista de la lógica de la misma.  2 - Incorporar Spring Security dentro de la aplicación 3 - Se modifican las relaciones en al menos una de las entidades propuestas para que los usuarios interactúen con el resto de entidades.  4 - Se añade una gestión por roles de las diferentes partes de la aplicación.  5 - Realizad las modificaciones oportunas del contenedor Docker.    Grupo 01: JavaBack  Arnau Prats Perelló  Federico Mathias Tito Hernández  Héctor Rubio Gil  Maria Sofía Ramis Company  Consultor: Josep Vañó Chic |

1. Modificar nuestro diagrama E/R para la incorporación de usuarios a nuestra aplicación tanto desde el punto de vista de la seguridad como desde el punto de vista de la lógica de la misma.

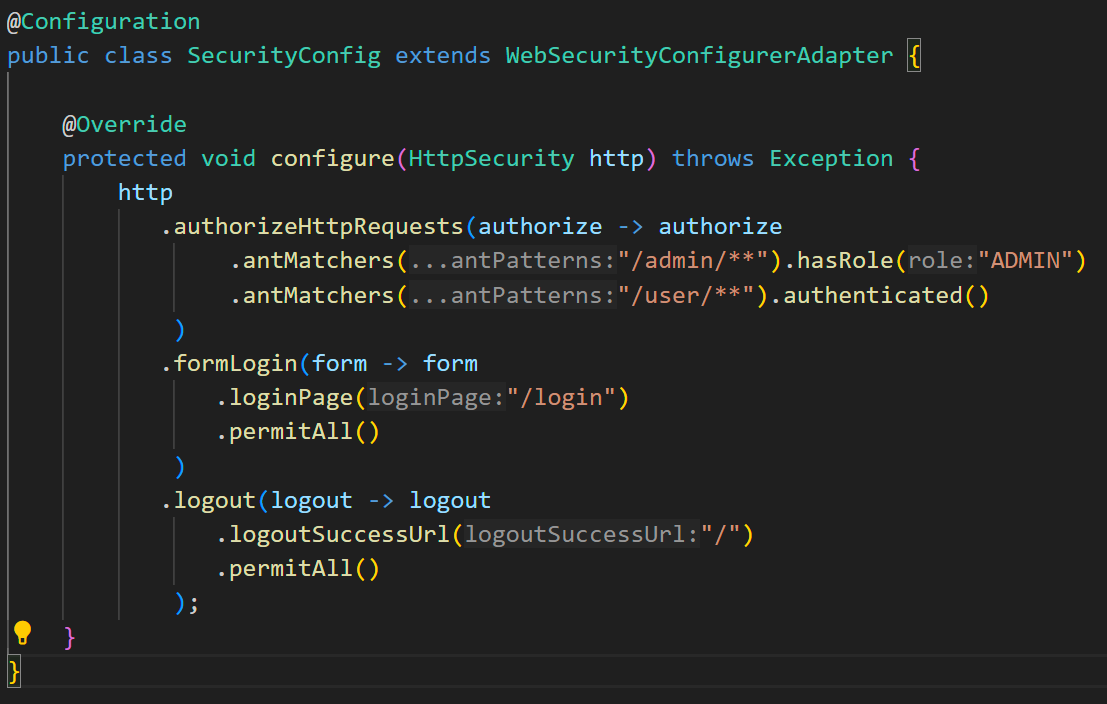


2 - Incorporar Spring Security dentro de la aplicación:

Para incluir Spring Security en un proyecto Java, se siguen unos pasos básicos, como es agregar dependencias de Spring Security en el pom.xml:

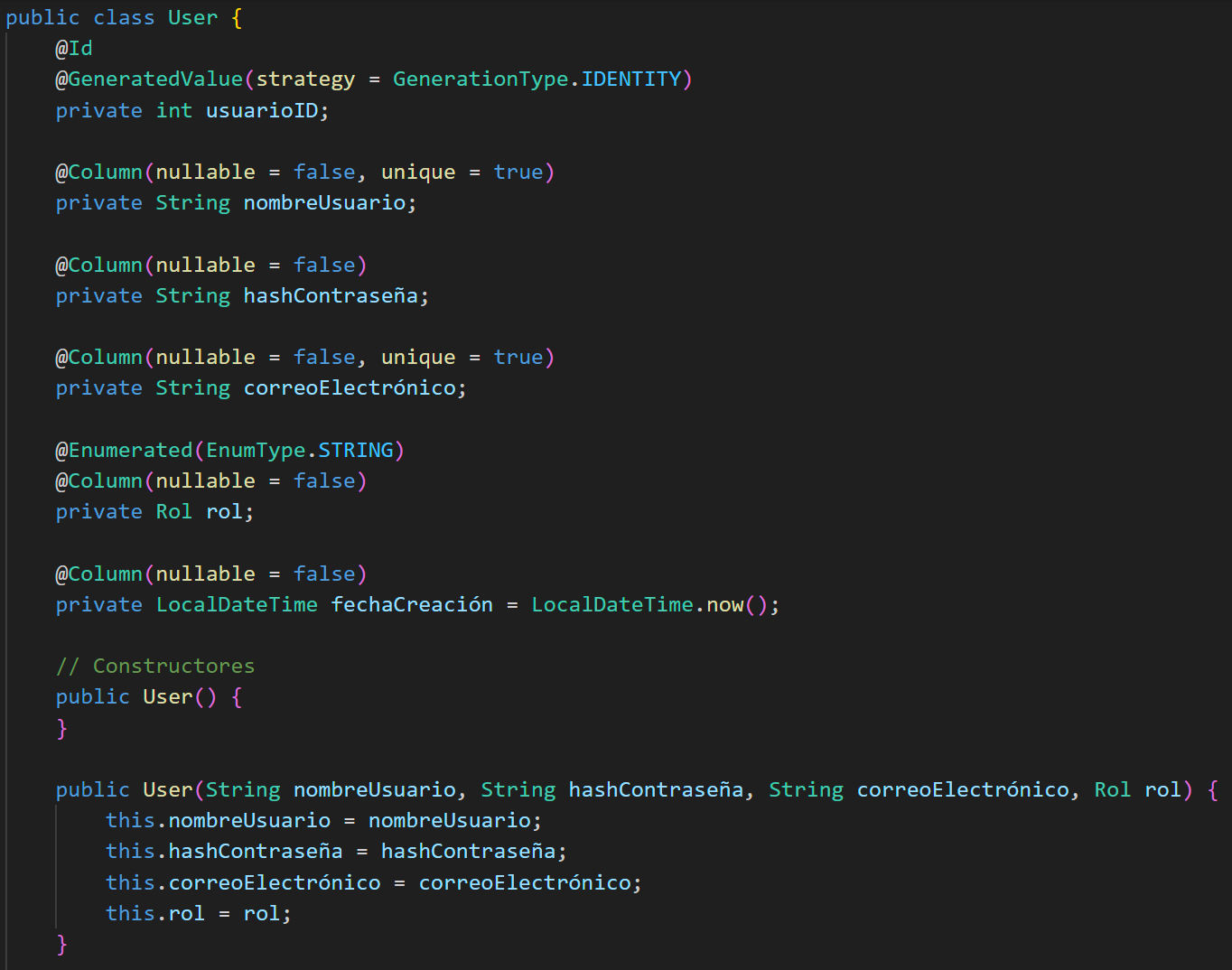


La creación de una clase Java que extienda WebSecurityConfigurerAdapter:

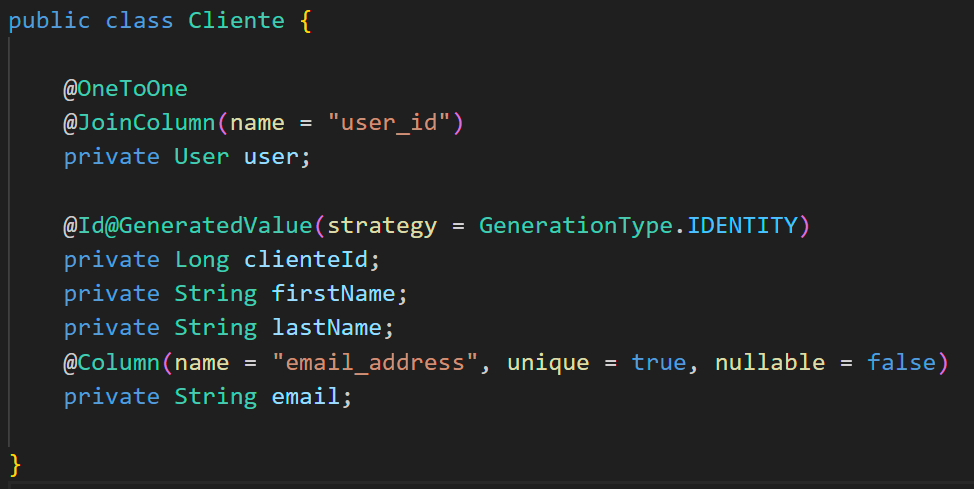


3 - Se modifican las relaciones en al menos una de las entidades propuestas para que los usuarios interactúen con el resto de entidades.:

Dado que estamos trabajando en una aplicación de alquiler de vehículos y queremos incorporar usuarios, vamos a crear el modelo para dicho usuario:



A continuación, vamos a modificar los modelos con las anotaciones de relación entre ellos. Aquí tenemos por ejemplo a la clase de clientes, que es 1 a 1 con el usuario:



Para finalizar dicho punto 3, se modificarán las vistas y los controladores para manejar la lógica de los nuevos alquileres y la interacción con los usuarios. Por ejemplo, en el controlador de alquileres, tendrás métodos para crear nuevos alquileres asociados a un usuario:

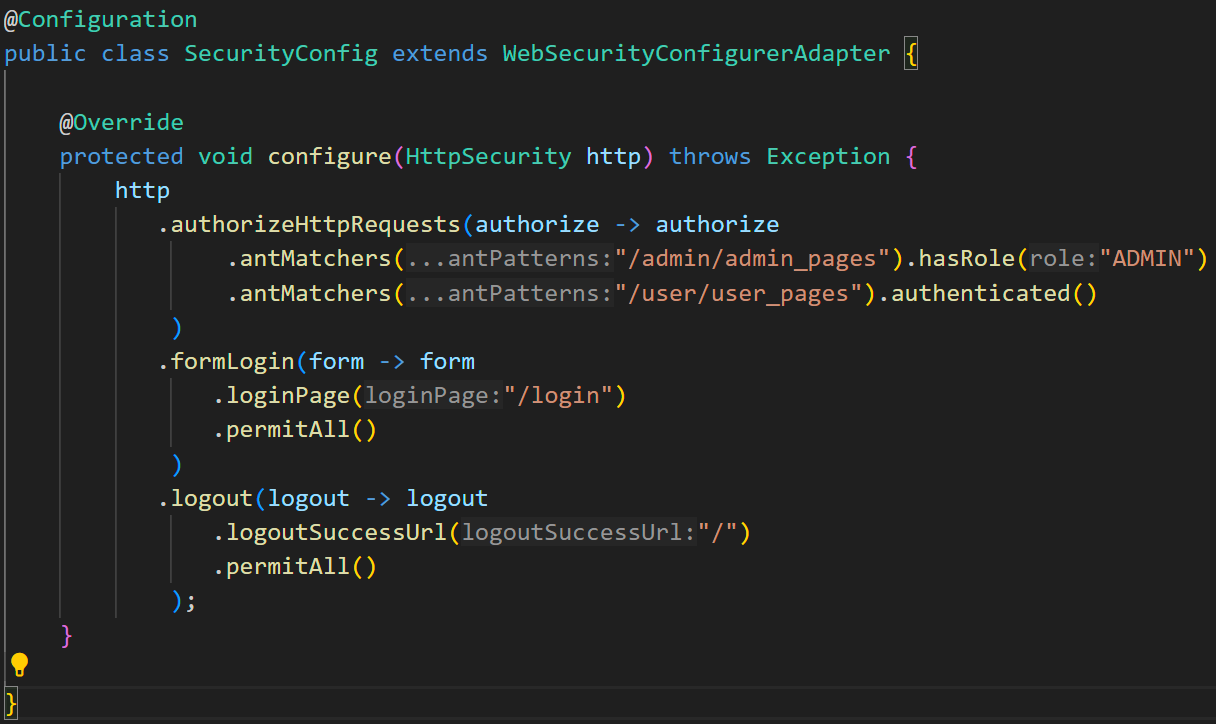


Y, por supuesto, modificar la vista de listado de alquileres:



4 – Se añade una gestión por roles de las diferentes partes de la aplicación:

Para que solo los usuarios con el rol ADMIN puedan acceder a los recursos bajo la ruta /admin/\*\*, la configuración en HttpSecurity usando la sintaxis lambda sería así:

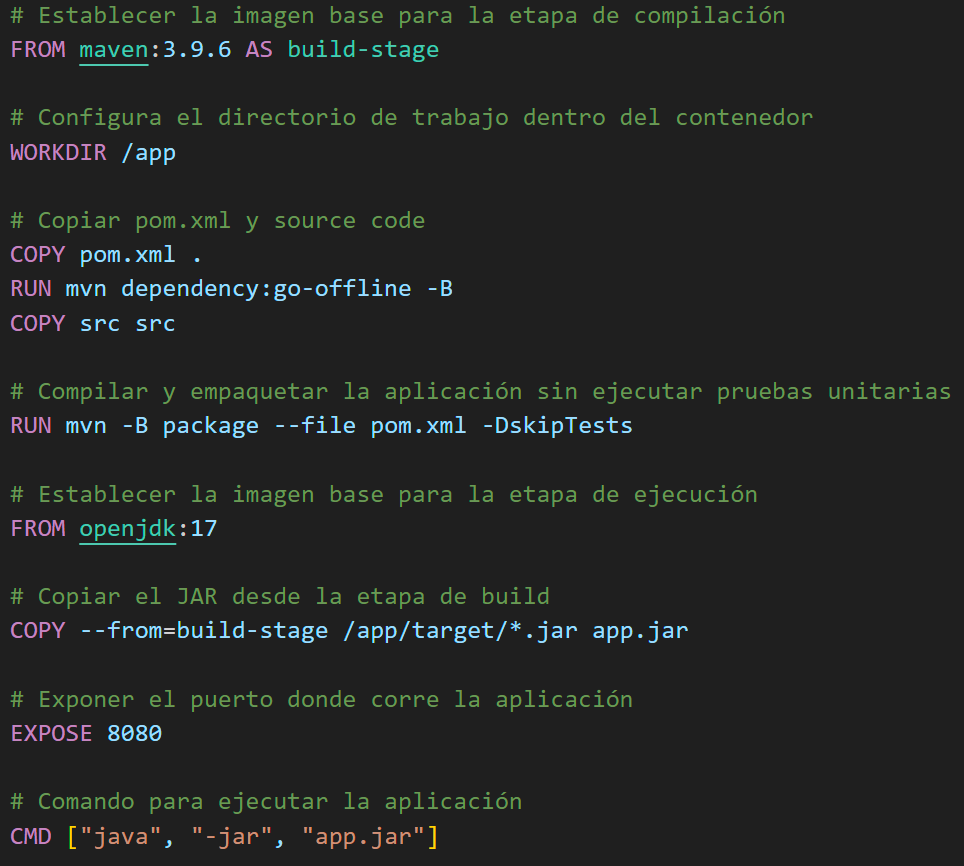


antMatchers("/admin/admin\_pages").hasRole("ADMIN") Asegura que solo los administradores puedan acceder a las rutas especificadas.

.antMatchers("/user/\*\*").authenticated(): Asegura que cualquier usuario autenticado pueda acceder a las rutas especificadas.

5 – Realizad las modificaciones oportunas del contenedor Docker:

Suponiendo que ya se tiene el Docker instalado y configurado de otras actividades, para hacer dicha creación del contenedor más sencilla desde Visual Studio Code, solo tenemos que crear un archivo dockerfile y rellenarlo con lo siguiente:



¡A partir de aquí, es necesario tener encendido el Docker!

docker build -t producto3\_javaback .

Este comando crea una imagen de docker a partir del Dockerfile que hemos creado antes, con el nombre producto1\_javaback.

docker run -p 8080:8080 producto3\_javaback

Este comando ejecuta un nuevo contenedor Docker utilizando la imagen producto3\_javaback, y hace que el puerto 8080 del contenedor esté disponible en el puerto 8080 del host local para recibir solicitudes.

Si todo sale satisfactoriamente, podemos ir a [localhost:8080/](localhost:8080/hello) y ver que nuestra actividad funciona correctamente.

6 – Generar el repositorio remoto en Github:

<https://github.com/ArnauPratsPerello/Producto3_JavaBack>