

Documento técnico

Contenido

Ide de desarrollo	2
Tecnología.....	2
Injector de dependencias	2
Control de versiones.....	2
Justificación del diseño.....	2
Spring Security.....	2
UserDetails	2
UserDetailsService.....	2
AuthenticationEntryPoint.....	2
OncePerRequestFilter.....	2
AuthenticationManager	2
PasswordEncoder	3
SecurityFilterChain	3
Models.....	3
Repository	3
Controller	3
Diagramas de secuencia	3
Prueba del proyecto	4
Requerimientos	4

Ide de desarrollo

Spring Tools for Eclipse 4.30.0.RELEASE

Tecnología

Spring Framework Boot 4.0.0-SNAPSHOT

Inyector de dependencias

Gradle 8.14.1

Control de versiones

<https://github.com/Lukisimas/DSSI>

Justificación del diseño

Para diseñar el back-end nos basamos en el MVC. Implementamos el Modelo y el Controlador. El diseño contempla la integración de la vista en caso de que se requiera trabajar el front-end.

Spring Security

Se utiliza el patrón de diseño de Spring Security. La clase que se encarga de implementar la seguridad es WebSecurityConfig que hace uso de las siguientes clases que deben ser implementadas.

UserDetails

Clase que contiene la información del usuario y los privilegios concedidos

UserDetailsService

Servicio que hace uso de UserDetails para buscar la información en el repositorio

AuthenticationEntryPoint

Componente que administra la autenticación y sus excepciones

OncePerRequestFilter

La clase AuthTokenFilter hereda de esta clase y se encarga de administrar los JSON Web Tokens que se devuelven en las Cookies.

AuthenticationManager

Esta clase es el puente entre el controlador AuthController y la seguridad de la aplicación

PasswordEncoder

Interfaz de servicio que especifica la forma en que se codifican las contraseñas.

SecurityFilterChain

Además, se requiere establecer un filtro de cadena. Esto es para interceptar las solicitudes que se hacen de HttpServletRequest y decidir si se aplican a lo especificado en la implementación del OncePerRequestFilter.

Models

Clases para establecer la información de los privilegios y los usuarios: TypeRole, Role y User.

Repository

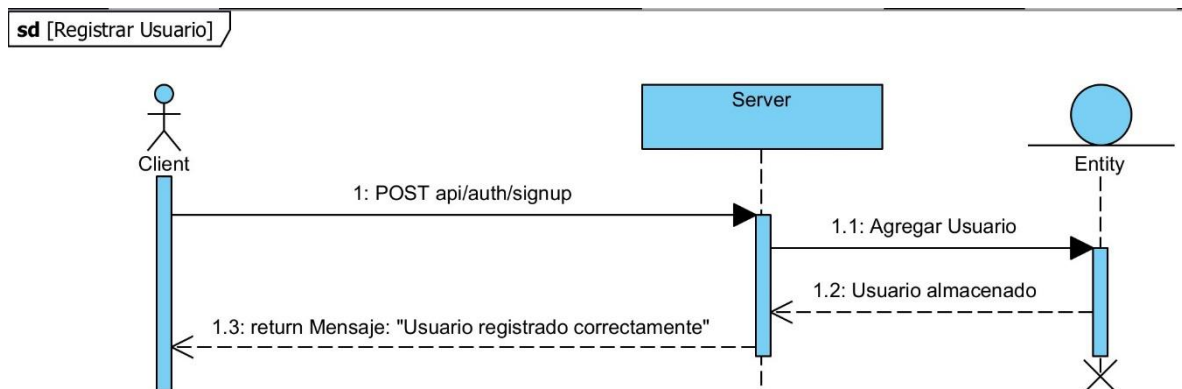
Clases que heredan de JpaRepository y hacen uso de las clases en el modelo. Se conectan a la base de datos para las operaciones de registro y consulta de la información usuarios.

Controller

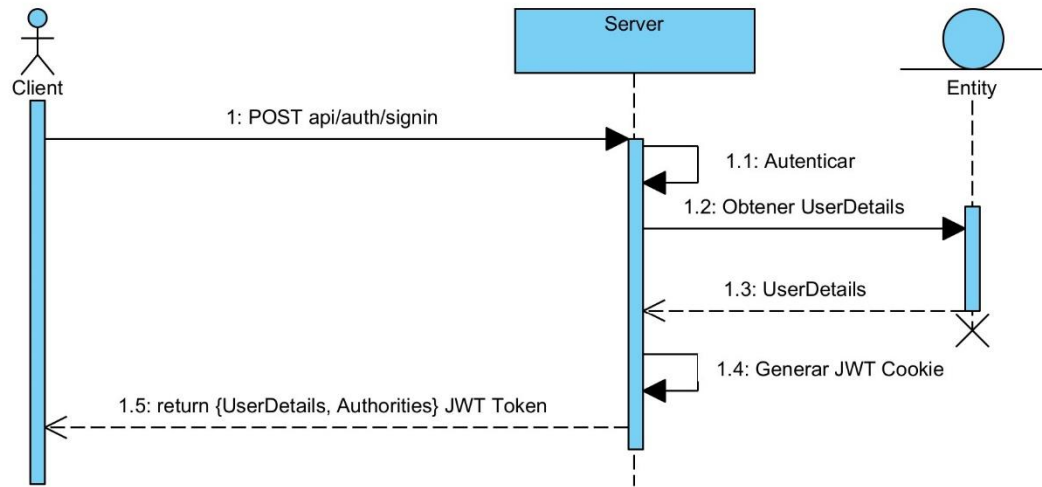
Contiene dos clases. AuthController se encarga de administrar los métodos post de signup, signin y signout en la ruta /api/auth.

TestController se encarga de administrar los métodos get de all, user, mod y admin en la ruta /api/test.

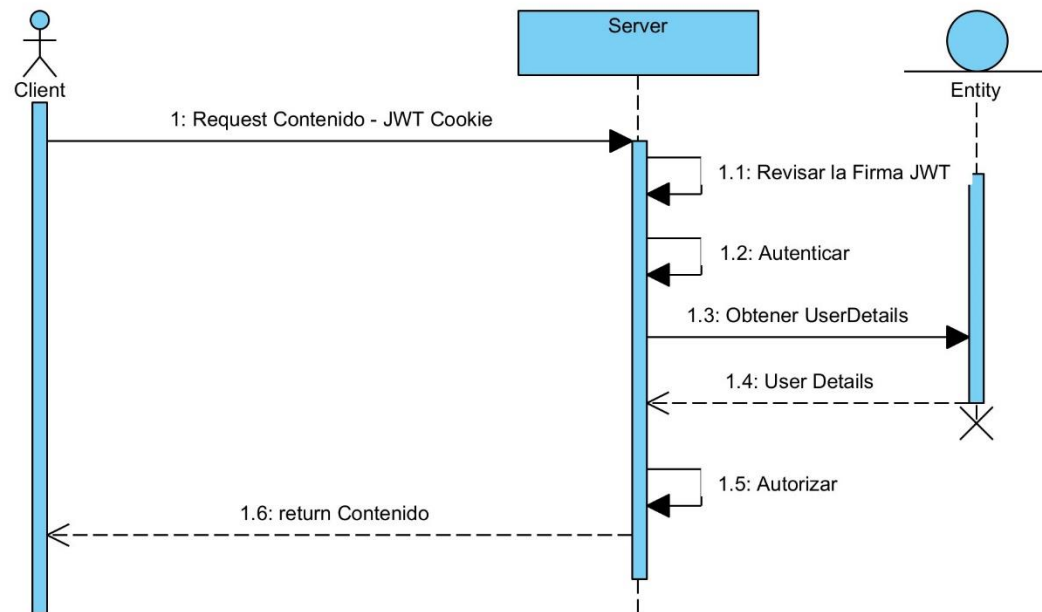
Diagramas de secuencia



sd [Autenticar Usuario]



sd [Acceder recurso]



Prueba del proyecto

Requerimientos

1. MySQL server 8 o superior. Revisar los datos de conexión en el archivo application.properties.
2. Puerto 8080 libre para levantar el servicio.
3. Ejecutar las consultas en la base de datos antes de hacer solicitudes.

```
INSERT INTO roles(name) VALUES('ROLE_USER');
```

```
INSERT INTO roles(name) VALUES('ROLE_MODERATOR');
```

```
INSERT INTO roles(name) VALUES('ROLE_ADMIN');
```

4. Se puede probar el back-end con herramientas como Rest Client y pasar la información del body request en formato JSON. Ejemplo para POST signup:

```
{"username":"alan","email":"alan.gil@unam.mx","role":["admin"],  
"password":"djhjf39472389"}
```