

Desafío - Sitio protegido con credenciales

- Para realizar este desafío debes haber estudiado previamente todo el material disponibilizado correspondiente a la unidad.
- Una vez terminado el desafío, comprime la carpeta que contiene el desarrollo de los requerimientos solicitados y sube el `.zip` en el LMS.
- Desarrollo desafío:
 - El desafío se debe desarrollar de manera Individual.

Descripción

Se necesita desarrollar una aplicación que posea dos (2) sitios de bienvenida, para distintos tipos de usuarios. El sistema debe permitir:

- El inicio de sesión de ambos usuarios con diferentes autoridades y tener la capacidad de prohibir el acceso a los recursos en base a la autoridad.
- Así mismo, asegurar la aplicación con *Spring Security* y utilizar la base de datos H2, además de escribir la consulta a la base de datos con *MyBatis*.

Requerimientos

1. Usar *Spring Initializr* para generar el proyecto con las respectivas dependencias.
2. Crear la enumeración *Role* en carpeta *model*.
ADMIN, CLIENT.
3. Crear la clase *Users* en carpeta *model* con los atributos:
 - *email*
 - *password*
 - *role de tipo Role*
4. Crear la interfaz *UsersMapper* en carpeta *mapper*.
5. Crear la clase *AuthService* en carpeta *service*.
6. Crear la clase *AdminController* en carpeta *controller*.

7. Crear la clase *ClientController* en carpeta *controller*.
8. Crear el template *admin.html* en *thymeleaf* dentro de *resources/templates*.
9. Crear el template *client.html* en *thymeleaf* dentro de *resources/templates*.
10. Crear la clase *WebSecurityConfig* en la carpeta *configuration*.
11. Crear la clase *LoginController* en carpeta.
12. Crear el template *login.html* en la carpeta *thymeleaf*.
13. Crear el controlador y template *403.html* de error personalizado para el recurso prohibido.
14. Permitir cerrar sesión apuntando a la ruta */logout*.
15. Agregar los datos de inicio en archivo *import.sql* y referenciar en *application.yml*.

Datos a cargar

```
create table users(email varchar, password varchar, role varchar);
-- vanrossum
insert into users(email, password, role)
values('guido@mail.com', '$2a$10$1X8qnW89UrB9zrjRvSLf4.H4u/ZkAXvyWoJMdXGw
TI1GA6t3vcquq', 'CLIENT');
-- gosling
insert into users(email, password, role)
values('james@mail.com',
'$2a$10$8bhmeA37YHbBaWpvFvSuC.3rtuZd.GV4fM63qSSoQ9SPSb4kelDze',
'ADMIN');
```

Archivo application.yml

```
spring:
  h2:
    console:
      enabled: false
  datasource:
    driver-class-name: org.h2.Driver
    schema: classpath:import.sql
    username: sa
    password: password
    url: jdbc:h2:mem:test
```