

DESARROLLO DE APLICACIONES WEB

LABORATORIO N° 14

Spring Boot Security



Alumno(s):	<ul style="list-style-type: none"> - <i>Núñez García Abraham Santiago</i> - <i>Perez Pomasoncco Sindell Geraldine</i> 				Nota	
Grupo:		Ciclo: IV				
Criterio de Evaluación	Excelente (4pts)	Bueno (3pts)	Requiere mejora (2pts)	No acept. (0pts)	Puntaje Logrado	
Identifica el modelo Spring Security						
Identifica la estructura de Spring Security						
Identifica las dependencias de Spring Security						
Conoce las capas de Spring Security						
Es puntual y redacta el informe adecuadamente						

Laboratorio 14: Spring Boot Security

Objetivos:

Al finalizar el laboratorio el estudiante será capaz de:

- Desarrollar una aplicación web empleando Spring Security

Seguridad:

- Ubicar maletines y/o mochilas en el gabinete del aula de Laboratorio.
- No ingresar con líquidos, ni comida al aula de Laboratorio.
- Al culminar la sesión de laboratorio apagar correctamente la computadora y la pantalla, y ordenar las sillas utilizadas.

Equipos y Materiales:

- Una computadora con:
 - Windows 10 o superior
 - Conexión a la red del laboratorio
- Instaladores:
 - Java SDK
 - Spring Tool Suite

Procedimiento y Resultados:

Aplicación

- Implemente una aplicación de registro y logueo de usuarios mediante Spring Boot Security

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

```
2024-06-10 15:03:07.948 INFO 25548 --- [ restartedMain] o.s.s.web.DefaultSe
2024-06-10 15:03:08.016 INFO 25548 --- [ restartedMain] o.s.b.w.embedded.tomcat
2024-06-10 15:03:08.024 INFO 25548 --- [ restartedMain] a.spring.Semana14A
$2a$10$DHje0Z81wee.oN2RFV1VNuzE8gxjMdD74D0IiRJ7HdE7WnFsz5IOe
$2a$10$EzxcRtw2gWmGNq3PDDzmeDH9Z8opD3c/V9Cwjb2BITrQtZ23MnI2
```

```

import javax.sql.DataSource;
@Configuration
public class ConfigSecurity extends WebSecurityConfigurerAdapter{
    @Autowired
    private DataSource ds;
    @Autowired
    private BCryptPasswordEncoder passEncripta;
    @Autowired
    public void autenticacion (AuthenticationManagerBuilder builder) {
        try {
            builder
                .jdbcAuthentication()
                .dataSource(ds)
                .passwordEncoder(passEncripta)
                .usersByUsernameQuery("select usuario, password, activo from us
                .authoritiesByUsernameQuery("select usuario, rol_nombre from ro
        } catch (Exception e){

    }
}

```

```

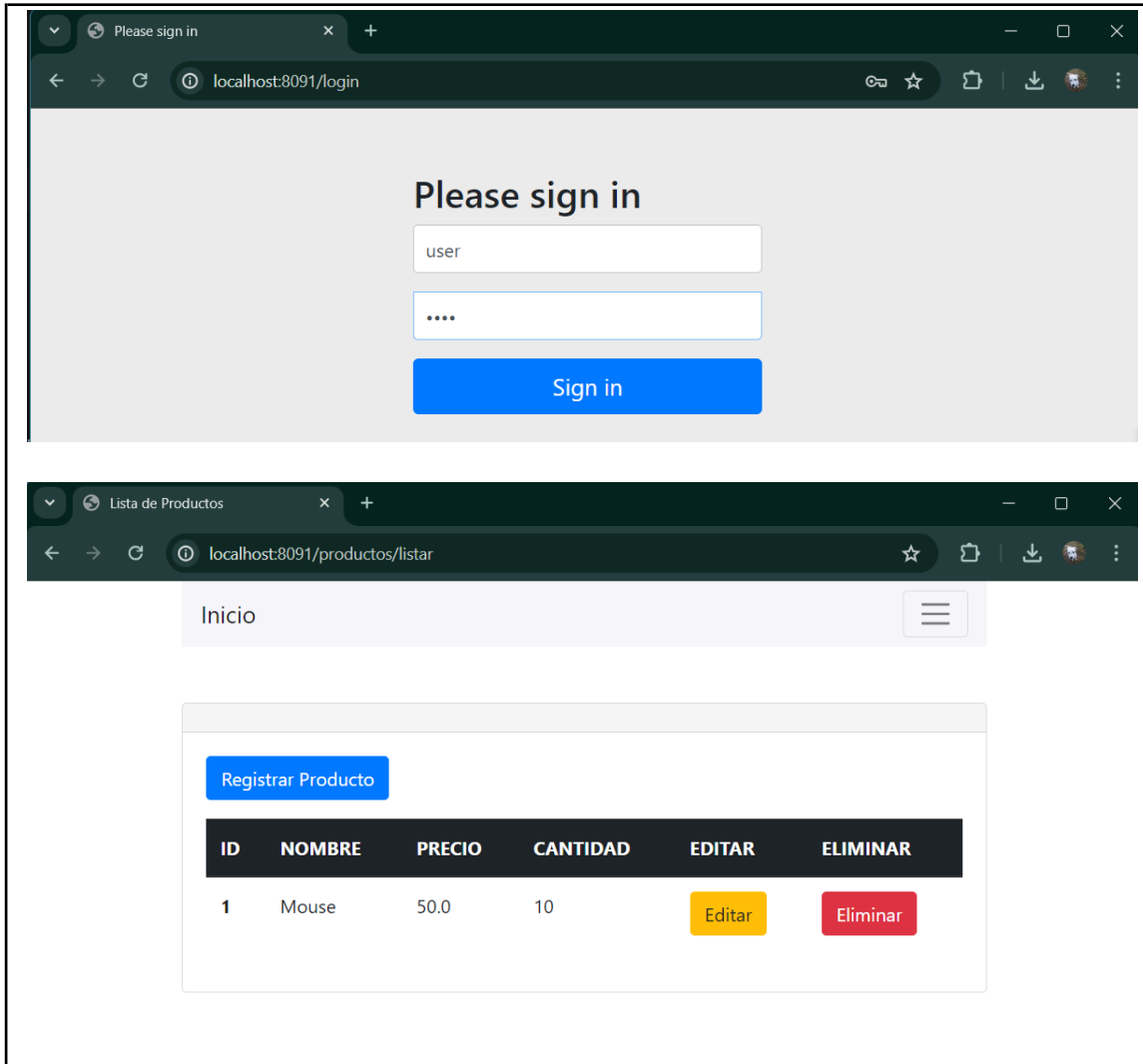
ConfigSecurity.java  Semana14AbApplication.java  Config.java  application.properties  Semana14AB/pom.xml
1 package aplicaciones.spring;
2 import org.springframework.beans.factory.annotation.Autowired;
3 import org.springframework.boot.CommandLineRunner;
4 import org.springframework.boot.SpringApplication;
5 import org.springframework.boot.autoconfigure.SpringBootApplication;
6 import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
7 @SpringBootApplication
8 public class Semana14AbApplication implements CommandLineRunner{
9
10     @Autowired
11     private BCryptPasswordEncoder passEncripta;
12
13     public static void main(String[] args) {
14         SpringApplication.run(Semana14AbApplication.class, args);
15     }
16
17     @Override
18     public void run(String... args) throws Exception {
19         String password1= "user";
20         String password2= "admin";
21         System.out.println(passEncripta.encode(password1));
22         System.out.println(passEncripta.encode(password2));
23     }
24 }

```

```

ConfigSecurity.java  Semana14AbApplication.java  Config.java  application.properties  Semana14AB/pom.xml
1 package aplicaciones.spring;
2
3 import org.springframework.context.annotation.Bean;
4 import org.springframework.context.annotation.Configuration;
5 import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
6 import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
7
8 @Configuration
9 public class Config implements WebMvcConfigurer{
10
11     @Bean
12     public BCryptPasswordEncoder encriptar() {
13         return new BCryptPasswordEncoder();
14     }
15 }
16

```



Conclusiones:

Indicar las conclusiones que llegó después de los temas tratados de manera práctica en este laboratorio.

1. En conclusión, Spring Security proporciona un marco integral y personalizable para manejar la autenticación y autorización en aplicaciones Spring Boot, permitiendo la implementación de mecanismos de seguridad robustos y flexibles con facilidad.
2. Para concluir, la integración de Spring Security con Spring Boot simplifica la configuración de la seguridad al ofrecer configuraciones predeterminadas sensatas y la posibilidad de personalización avanzada, lo que facilita la protección de aplicaciones contra amenazas comunes.
3. En resumen, el uso de Spring Security en Spring Boot no solo mejora la seguridad de las aplicaciones web, sino que también agiliza el desarrollo mediante la integración de características como la protección contra ataques CSRF y la gestión de sesiones, contribuyendo a un entorno de desarrollo más seguro y eficiente.

