**Medidas de Seguridad con Spring Security**

**Spring Security** es un framework de seguridad ampliamente utilizado en aplicaciones Spring para manejar la autenticación y autorización de usuarios.

Para agregar Spring Security a un proyecto, necesitas incluir la dependencia **spring-boot-starter-security**. Esta dependencia proporciona las funcionalidades básicas para la autenticación y autorización en aplicaciones Spring Boot.

Código:

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

Se necesita realizar pruebas de seguridad, puedes incluir la dependencia **spring-security-test**:

|  |
| --- |
| <dependency>  <groupId>org.springframework.security</groupId>  <artifactId>spring-security-test</artifactId>  <scope>test</scope>  </dependency> |

Las dependencias permitirán configurar y utilizar las características de seguridad de Spring Security dentro del proyecto.

**Ejemplificacion:**

Nuestro primer acceso como usuarios a una aplicación consta en la verificación de nuestra información relacionada a esta, siendo este proceso de autenticación un paso crucial correspondiente a la seguridad en un sistema. Spring Security nos brinda varias estrategias para la implementación de un sistema compuesto por una autenticación.

Spring Security nos brinda una implementación convencional para una autenticación básica por medio de la autenticación por formularios, estando estructurada por ejemplo de la forma:

|  |
| --- |
| @Configuration  @Enablewebsecurity  public class SecurityConfig extends WebSecurityConfigurerAdapter {    @Autowired  public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {  auth  .inkenoryAuthentication()  .withUser("usuario").password("contraseña").roles("PRAGMATICO");  }  @Override  protected void configure(HttpSecurity http) throws Exception {  http  .authorizeRequests()  .antMatchers("/publico/\*\*").permitAll()  .anyRequest().authenticated()  .and()  .formLogin()  .loginPage("/login").permitAll()  .and()  .logout()  .permitAll()  }  } |

En nuestro código, tenemos la autenticación por medio de usuario y contraseña para el rol de pragmático, para lo cual establecemos permisos dependiendo del estado de la autenticación por medio de las rutas ‘/publico/\*\*’ y ‘/login’ siendo esta última la accedida una vez se completa el proceso de autenticación.

Prevención de ataques:

Spring Security también cuenta con mecanismos para prevenir ataques denominados como ataques de fuerza bruta, los cuales corresponden a la repetición de intentos de inicios de sesión con el fin de dar con la contraseña correcta del usuario al cual buscan vulnerar, empleando herramientas como CAPTCHAs y bloqueos temporales debido a múltiples intentos fallidos.

Por otro lado, Spring Security nos da acceso a la integración de autenticadores externos, como Google o GitHub, mediante protocolos de autenticación federada, con OAuth 2.0. Una vez la autenticación externa es completada exitosamente, nos redirige nuevamente a nuestra aplicación junto con un token de acceso el cual proporciona la información del usuario por medio del método ‘oauth2Login()’.

|  |
| --- |
| @Configuration  @EnableWebSecurity  public class SecurityConfig extends WebSecurityConfigurerAdapter {  @Override  protected void configure(HttpSecurity http) throws Exception \{  http  .authorizeRequests()  .antMatchers("/").permitAll()  .anyRequest().authenticated()  .and()  .oauth2Login()  .loginPage("/login")  .defaultSuccessUr1("/home")  .userInfoEndpoint()  .userService(oAuth2UserService());  }  @Bean  public OAuth2UserService<OAuth2UserRequest, OAuth2User> OAuth2UserService() {  return new DefaultOAuth2UserService();  }  } |

Dicha autenticación mediante tokens nos da ventajas significativas como que los tokens generados no se almacenan en el servidor. Así, una vez este haya vencido, por ejemplo, por un tiempo de validez establecido, se debe realizar de nuevo el proceso de autenticación permitiéndonos manejar cada solicitud de manera independiente.

**Spring. Módulos y Dependencias del Proyecto** **Retrieved April 1, 2025 from:**

<https://docs.spring.io/spring-security/reference/modules.html>

Spring. Security with Spring Series (January 18, 2024) from:

<https://www.baeldung.com/security-spring>

**Spring. Autenticación y Autorización: Medidas de Seguridad con Spring Security Feb 15, 2024 from:**

<https://medium.com/somos-pragma/autenticaci%C3%B3n-y-autorizaci%C3%B3n-medidas-de-seguridad-con-spring-security-3f3de4e0b1b2>