

Cuarta parte

Aplicar seguridad a las aplicaciones web



Aplicando seguridad a nuestras aplicaciones web

En este curso aprenderás a aplicar 2 tipos de seguridad a tus aplicaciones web:

- **Seguridad implementada con Spring Security.**
 - ✓ **Sección titulada:** “Implementar seguridad con Spring Security”.
- **Seguridad implementada con el estándar de Java EE.**
 - ✓ **Sección titulada:** “Implementar seguridad con el estándar de Java EE - JDBCRealm”.
 - Veremos este tipo de seguridad porque es muy usada en otro tipo de aplicaciones (no desarrolladas con Spring).

Nota:

En las 2 secciones dedicadas a la seguridad, continuaremos con el proyecto terminado hasta la sección anterior: “Integración de Spring MVC y Spring Data JPA”.

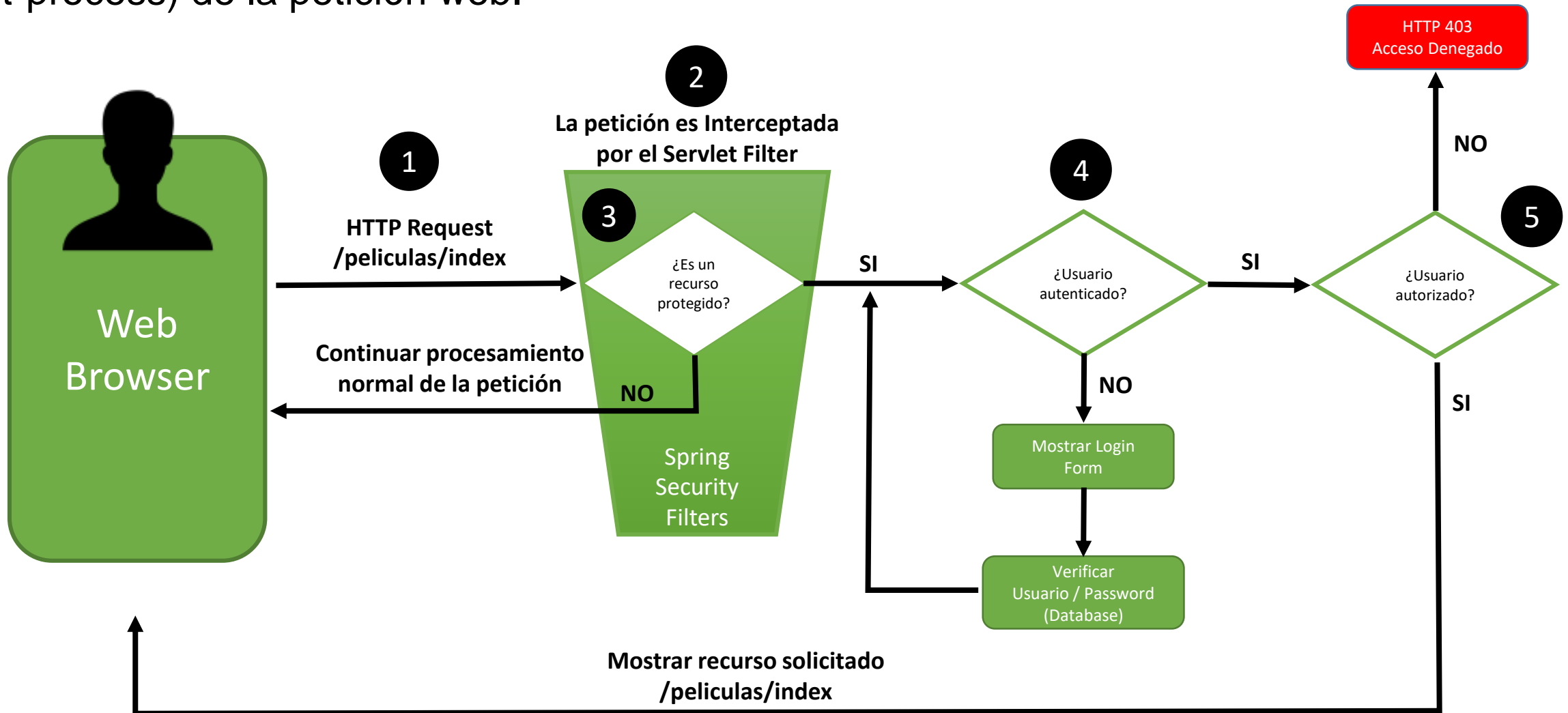
- ✓ Estas dos secciones dedicadas a la seguridad no llevan secuencia. Puedes ver las siguientes 2 secciones en el orden que tu desees.
- ✓ En ambas secciones continuaremos con el proyecto descargado de la lección anterior:
cineapp-sin-seguridad.zip → Proyecto terminado con toda la funcionalidad, **PERO SIN SEGURIDAD**.

¿Qué es Spring Security?

- Es un framework de seguridad (módulo de Spring) que permite aplicar seguridad a tus aplicaciones desarrolladas con Spring.
 - ✓ En este curso veremos como aplicar **Seguridad en Aplicaciones Web**.
- Spring Security aplica 2 tipos de seguridad:
 - ✓ **Autenticación**: ¿Es un usuario válido para acceder a la aplicación?
 - ✓ **Autorización**: ¿El usuario tiene permisos (ROL) para acceder al recurso solicitado?
- La seguridad es aplicada a nivel de **Petición Web (HTTP Request)** y a nivel de **Invocación de Métodos**.
- Spring Security esta basado en Spring Framework. Internamente utiliza:
 - ✓ Inyección de Dependencias (DI)
 - ✓ Programación orientada a aspectos (AOP).
- En aplicaciones web Spring Security utiliza **Servlet Filters** para aplicar seguridad a las peticiones web y restringir el acceso a nivel de URL.

Spring Security – Servlet Filter

- Spring Security utiliza varios Servlet Filters para filtrar las peticiones web.
- Los Servlet Filters son componentes (Interceptors) ejecutados antes (pre-process) y después (post-process) de la petición web.



Dependencias - Spring Security (pom.xml)

```
<dependency>
  <groupId>org.springframework.security</groupId>
  <artifactId>spring-security-web</artifactId>
  <version>5.2.0.RELEASE</version>
</dependency>

<dependency>
  <groupId>org.springframework.security</groupId>
  <artifactId>spring-security-config</artifactId>
  <version>5.2.0.RELEASE</version>
</dependency>

<dependency>
  <groupId>org.springframework.security</groupId>
  <artifactId>spring-security-taglibs</artifactId>
  <version>5.2.0.RELEASE</version>
</dependency>

<!-- (you don't need this if you are using a .RELEASE version) -->
<repositories>
  <repository>
    <id>spring-milestones</id>
    <name>Spring Milestones</name>
    <url>https://repo.spring.io/libs-milestone</url>
    <snapshots>
      <enabled>false</enabled>
    </snapshots>
  </repository>
</repositories>
```

Spring Security

**Spring Security
Tag (JSPs)**

Spring Security
<https://spring.io/projects/spring-security>

Configuración de Spring Security (1)

➤ Servlet Filter (web.xml)

```
<!-- Spring Security Filter -->
<filter>
  <filter-name>springSecurityFilterChain</filter-name>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
<filter-mapping>
  <filter-name>springSecurityFilterChain</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```



Todas las URLs (HTTP Requests) de la aplicación serán interceptadas (FILTRADAS) por Spring Security, antes de ser procesadas.

Configuración de Spring Security (2)

- Especificar archivo XML con la configuración de Spring Security (web.xml)

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    /WEB-INF/spring/root-context.xml,
    /WEB-INF/spring/security.xml
  </param-value>
</context-param>
```

Archivo XML con la configuración de Spring Security:

- ✓ Usuarios
- ✓ Roles
- ✓ Recursos protegidos
- ✓ Ruta del formulario de login.
- ✓ Etc.

Configuración de Spring Security (3)

➤ Configuración MÍNIMA de Spring Security (/WEB-INF/spring/security.xml)

```
<?xml version="1.0" encoding="UTF-8"?>

<b:beans xmlns="http://www.springframework.org/schema/security"
  xmlns:b="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/security
    http://www.springframework.org/schema/security/spring-security.xsd">

  <http />

  <authentication-manager>

    <authentication-provider>
      <user-service>
        <user name="luis" password="{noop}luis123" authorities="EDITOR" />
        <user name="marisol" password="{noop}maril23" authorities="GERENTE" />
      </user-service>
    </authentication-provider>

  </authentication-manager>

</b:beans>
```

Con esta configuración:

- ❖ Será requerida autenticación para todas las URLs.
- ❖ Spring generará un formulario HTML de login de forma automática.
- ❖ Se crearán 2 usuarios en memoria con los roles especificados.
- ❖ Se agregará CSRF attack prevention (Cross-site request forgery).

Spring Security – Especificar autorización por ROL.

➤ Los tags `<intercept-url>` son declarados dentro del tag `<http>` y son utilizados para definir conjuntos de URLs que estarán protegidas en la aplicación. Los atributos más usados son:

✓ **pattern**: sirve para indicar un patrón de URLs. Ejemplo:

- **/peliculas/***: Todas las URLs que comiencen con **/peliculas/** (/peliculas/index, /peliculas/create, /peliculas/save, etc.)

✓ **access**: especificar atributos de acceso: Ejemplo:

- **access="hasAnyAuthority('EDITOR')"**: Permitir el acceso **ÚNICAMENTE** a usuarios con el **ROL EDITOR**.

```
<?xml version="1.0" encoding="UTF-8"?>

<b:beans xmlns="http://www.springframework.org/schema/security"
  xmlns:b="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/security
    http://www.springframework.org/schema/security/spring-security.xsd">

  <http auto-config="true">
    <!-- Declaramos todos los recursos que estaran protegidos -->
    <intercept-url pattern="/peliculas/*" access="hasAnyAuthority('EDITOR') " />
    <intercept-url pattern="/horarios/*" access="hasAnyAuthority('EDITOR') " />
    <intercept-url pattern="/noticias/*" access="hasAnyAuthority('EDITOR') " />
    <intercept-url pattern="/banners/*" access="hasAnyAuthority('GERENTE') " />
  </http>
  . . .
</b:beans>
```

Controlador para cerrar la sesión

El objeto request es necesario para que Spring obtenga la sesión actual para invalidarla.

Implementación de Spring Security encargada de destruir la sesión.

Controller

```
@GetMapping(value="/logout")
public String logout(HttpServletRequest request){

    SecurityContextLogoutHandler logoutHandler =
    new SecurityContextLogoutHandler();

    logoutHandler.logout(request, null, null);

    return "redirect:/login";
}
```

Después de cerrar sesión redireccionamos al usuario al formulario de login.

Method Detail

logout

```
public void logout(javax.servlet.http.HttpServletRequest request,
    javax.servlet.http.HttpServletResponse response,
    Authentication authentication)
```

Requires the request to be passed in.

Specified by:

logout in interface LogoutHandler

Parameters:

request - from which to obtain a HTTP session (cannot be null)

response - not used (can be null)

authentication - not used (can be null)

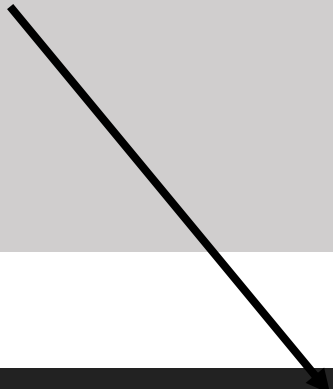
Documentación Oficial.

<https://docs.spring.io/spring-security/site/docs/5.0.0.RC1/api/org/springframework/security/web/authentication/logout/SecurityContextLogoutHandler.html>

Link para cerrar la sesión

menu.jsp

```
<div id="navbar" class="navbar-collapse collapse">
  <ul class="nav navbar-nav">
    <li><a href="/peliculas/indexPaginate?page=0">Peliculas</a></li>
    <li><a href="/horarios/indexPaginate?page=0">Horarios</a></li>
    <li><a href="/noticias/index">Noticias</a></li>
    <li><a href="/banners/index">Banner</a></li>
    <li><a href="/logout">Salir</a></li>
  </ul>
</div>
```



My CineSite Peliculas Horarios Noticias Banner Salir