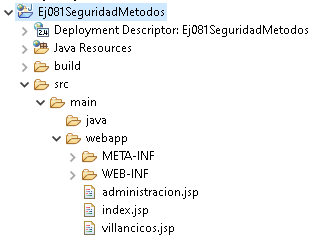
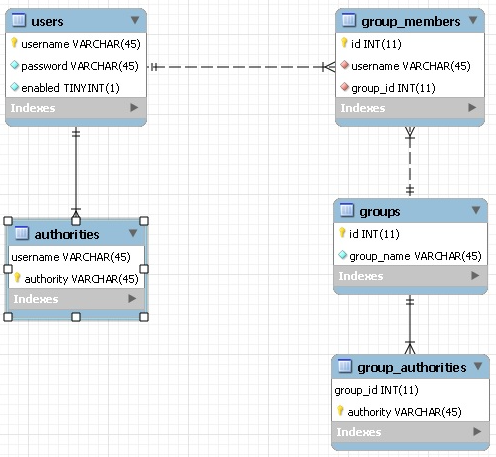
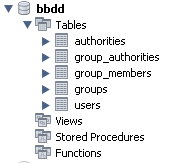
**Proyecto 5 Pérdida de Control de Acceso**

Vamos a segurizar las vistas jsp del proyecto.



Debemos tener la base de datos MySQL instalada.



**Descriptor de despliegue web.xml de la carpeta WEB-INF**

El descriptor de despliegue (web.xml) es un archivo fundamental en las aplicaciones web Java EE que sirve como configurador central de la aplicación. Veamos sus funciones principales:

1. Configuración de Componentes

* Define la estructura y organización de la aplicación web
* Especifica cómo se deben desplegar los diferentes componentes
* Establece las relaciones entre los elementos de la aplicación

2. Mapeo de URLs

* Asigna patrones de URL a servlets específicos
* Define cómo se deben manejar las diferentes rutas de la aplicación
* Permite configurar múltiples servlets y sus respectivos mapeos

3. Seguridad y Autenticación

* Configura los filtros de seguridad
* Define los roles y permisos de acceso
* Establece las restricciones de seguridad para diferentes recursos

4. Gestión de Recursos

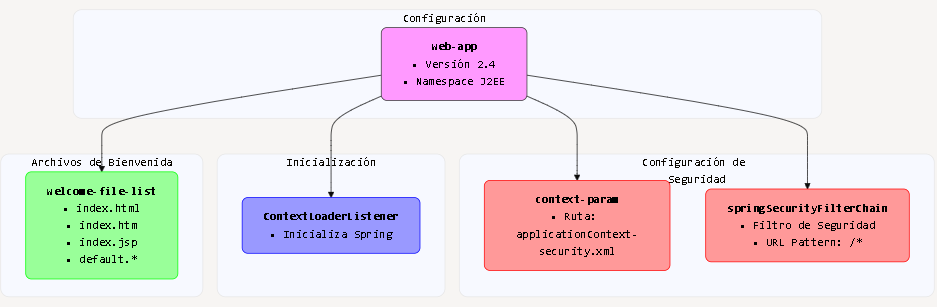
* Configura los archivos de bienvenida (welcome files)
* Define los parámetros de contexto de la aplicación
* Establece las referencias a recursos externos

5. Inicialización

* Configura los listeners de la aplicación
* Define el orden de inicialización de componentes
* Establece los parámetros de inicio de la aplicación

6. Integración con Frameworks

* Permite la integración con frameworks como Spring
* Configura los filtros y listeners específicos del framework
* Define las rutas de configuración del framework



1. Declaración XML y Namespace

* Especifica la versión XML 1.0 y codificación UTF-8
* Define el esquema XML para validación docs.oracle.com
* Establece la versión 2.4 del servlet, que permite características modernas como anotaciones

2. Configuración de Seguridad Spring

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/applicationContext-security.xml</param-value>

</context-param>

* Indica la ubicación del archivo de configuración de seguridad de Spring
* Este archivo contiene la configuración detallada de la seguridad de la aplicación

3. Filtro de Seguridad Spring

<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>

* Configura el filtro principal de Spring Security
* El patrón /\* significa que el filtro interceptará todas las solicitudes cloud.google.com
* El DelegatingFilterProxy actúa como intermediario entre el contenedor servlet y los filtros de Spring Security

4. Listener de Contexto

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

* Inicializa el contexto de Spring en el arranque de la aplicación
* Esencial para la configuración y funcionamiento de Spring Security.

5. Archivos de Bienvenida

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

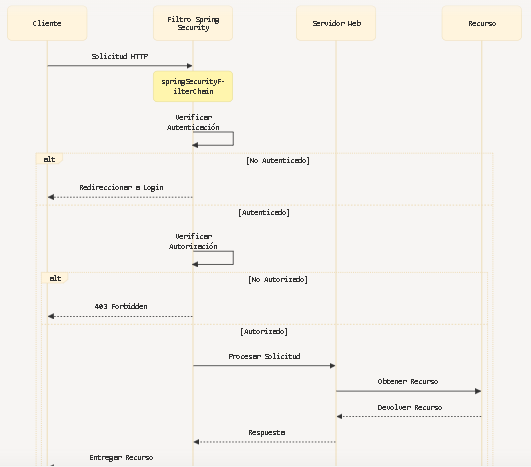
<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

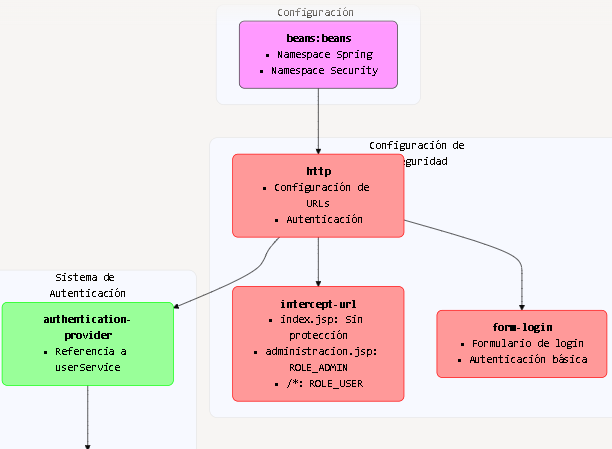
* Especifica los archivos que se mostrarán cuando se acceda a una carpeta sin especificar un archivo
* El servidor intentará cada archivo en orden hasta encontrar uno disponible

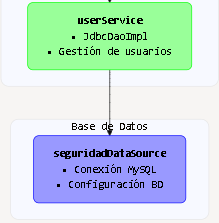
Flujo de Seguridad



**Archivo de configuración principal de Spring Security applicationContext-security.xml**

El archivo applicationContext-security.xml es el archivo de configuración principal de Spring Security que define cómo se manejará la seguridad en nuestra aplicación. Veamos su estructura y funcionamiento:





1. Declaración XML y Namespace

<beans:beans xmlns="http://www.springframework.org/schema/security"

xmlns:beans="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:p="http://www.springframework.org/schema/p"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.0.xsd

http://www.springframework.org/schema/security http://www.springframework.org/schema/security/spring-security-2.0.1.xsd">

* Define los espacios de nombres necesarios para Spring Security
* Establece la estructura XML para la configuración de seguridad
* Permite la validación del archivo contra los esquemas de Spring

2. Configuración de URLs y Acceso

<http>

<intercept-url pattern="/index.jsp" filters="none" />

<intercept-url pattern="/administracion.jsp" access="ROLE\_ADMIN" />

<intercept-url pattern="/\*" access="ROLE\_USER" />

<form-login />

<anonymous />

<http-basic />

<logout logout-success-url="/index.jsp" />

</http>

* Define las reglas de acceso para diferentes URLs
* Establece tres niveles de acceso:
* /index.jsp: Acceso público sin restricciones
* /administracion.jsp: Solo para usuarios con rol ADMIN
* Todas las demás URLs: Requieren rol USER
* Configura el sistema de autenticación con formulario de login
* Permite acceso anónimo y autenticación básica HTTP
* Define la URL de redirección después del logout

3. Proveedor de Autenticación

<authentication-provider user-service-ref="userService" />

<beans:bean id="userService" class="org.springframework.security.userdetails.jdbc.JdbcDaoImpl">

<beans:property name="dataSource" ref="seguridadDataSource" />

<beans:property name="enableGroups" value="true" />

<beans:property name="enableAuthorities" value="false" />

</beans:bean>

* Configura el servicio de autenticación usando una base de datos
* Utiliza JdbcDaoImpl para la gestión de usuarios
* Habilita el soporte para grupos pero deshabilita autoridades individuales
* Conecta con la fuente de datos configurada

4. Configuración de la Base de Datos

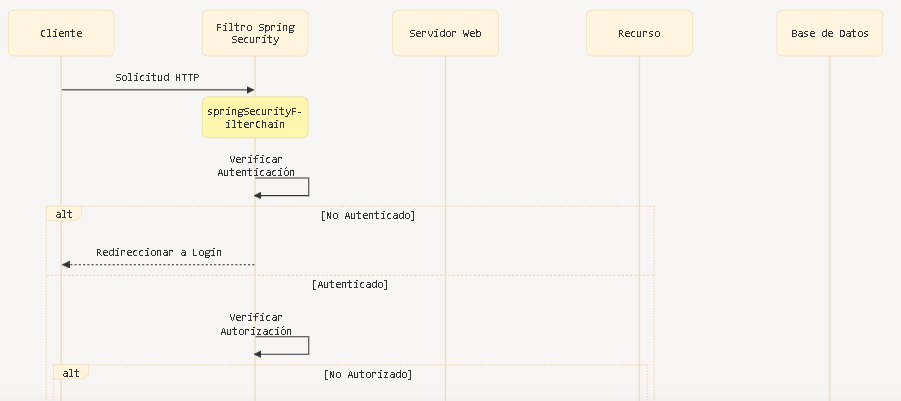
<beans:bean id="seguridadDataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close"

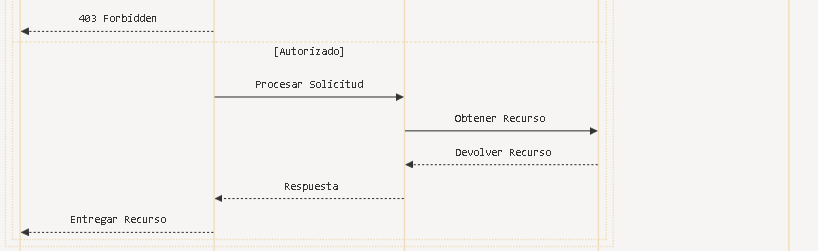
p:driverClassName="com.mysql.jdbc.Driver" p:url="jdbc:mysql://localhost:3306/bbdd"

p:username="root" p:password=""/>

* Configura la conexión a la base de datos MySQL
* Utiliza el pool de conexiones DBCP para mejor rendimiento
* Define los parámetros de conexión:
* Driver de MySQL
* URL de la base de datos
* Credenciales de acceso

**Flujo de Seguridad**





Proceso de Autenticación:

* Cuando un cliente hace una solicitud, el filtro springSecurityFilterChain la intercepta
* Si el usuario no está autenticado, se redirige al formulario de login configurado en <form-login />
* La autenticación se realiza contra la base de datos configurada en seguridadDataSource

Proceso de Autorización:

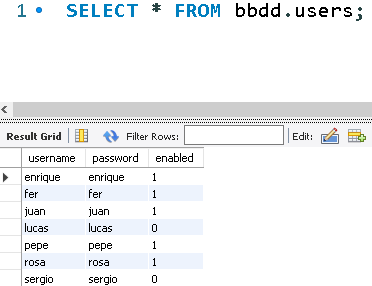
* Una vez autenticado, el sistema verifica los permisos según las reglas definidas en <intercept-url>
* Por ejemplo, si se intenta acceder a /administracion.jsp, se verifica que el usuario tenga el rol ADMIN
* Si no tiene los permisos necesarios, se devuelve un error 403 (Forbidden)

Acceso a Recursos:

* Si la autenticación y autorización son exitosas, la solicitud llega al servidor web
* El recurso solicitado se procesa y se devuelve al cliente
* Todo el proceso está protegido por el filtro de seguridad configurado

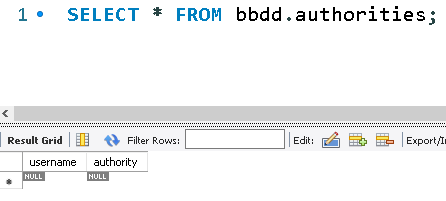
**En la base de datos**

Tabla users



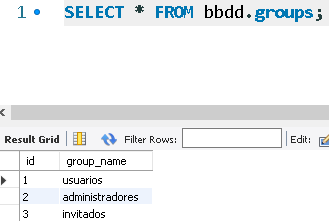
El campo ***enabled*** nos permite habilitar o deshabilitar usuarios. Puedo deshabilitar un usuario cuando esté baneado por mala conducta, o puede ser que esté intentando hackear lo que tengo o que ha entrado desde otro dispositivo que no es el correcto o se alcanzó el máximo de intentos de login permitidos, entre otras causas.

Tabla authorities



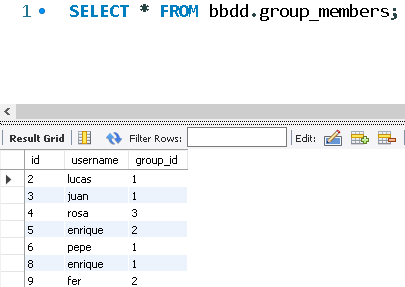
Un ejemplo de tupla de valores podría ser: Juan – ROLE\_ADMIN. Me sirve si quisiera autorizar solo mediante el sistema de roles. Sin embargo, en este ejercicio hacemos autenticación y autorización basado en grupos.

Tabla groups



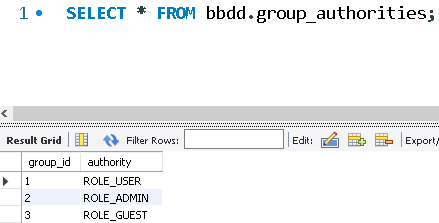
El grupo de invitados no tiene asignado ningún rol.

Tabla group\_members



Rosa es invitada, Lucas es usuario pero Enrique es usuario y administrador.

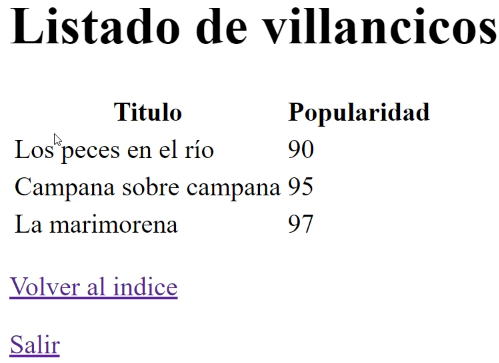
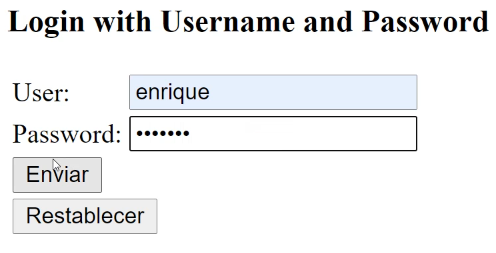
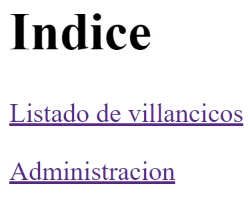
Tabla group\_authorities

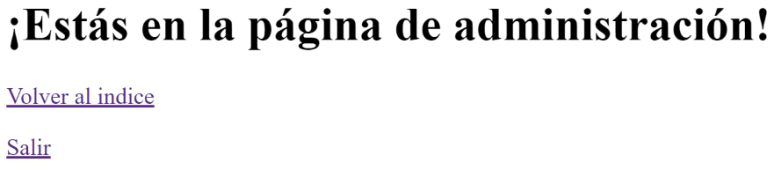


Permite asignar roles a los diferentes grupos.

**Testing Funcional de la Aplicación**

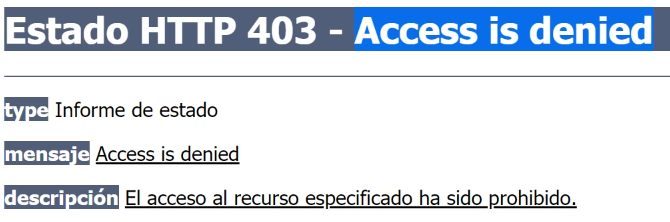
Caso 1) Rol usuario y administrador: aquí pueden acceder los usuarios y administradores. Como Enrique posee ambos roles, entonces puede ingresar, no solo a villancicos sino también a administración.



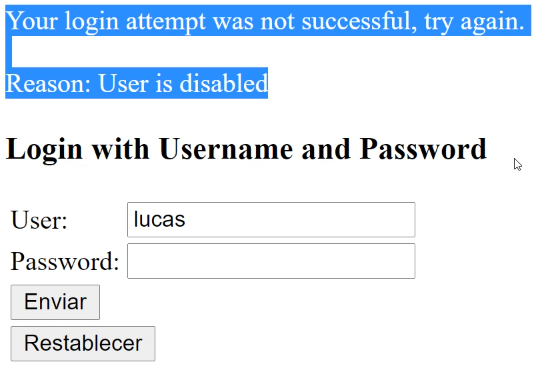
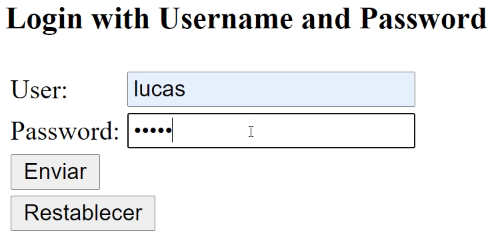


El enlace de Salir, ejecuta un Logout y quita al usuario anterior. Por lo tanto, si luego de salir intento ingresar a administración por ejemplo ya no me va a dejar hacerlo.

Caso 2) Rol administrador: el usuario Fer solo puede ingresar a administración pero no a villancicos. En esta ultima situación, aparece el mensaje de denegación. Debo hacer un Logout al final para quitar al último usuario.



Caso 3) usuario deshabilitado. Lucas no puede acceder por estar deshabilitado.



Nos avisara la aplicación que el usuario esta deshabilitado.