

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)

NUEVO ¿Quieres saber cuánto ganas en relación al mercado? pincha aquí...

[Ver cursos que ofrece Autentia](#)

[Descargar comics en PDF y alta resolución](#)



[iNUEVO!] 2008-12-01

2008-11-17

2008-09-01

2008-07-31

Estamos escribiendo un libro sobre la profesión informática y estas viñetas formarán parte de él. Puedes opinar en la sección [comic](#).

Catálogo de servicios Autentia (PDF 6,2MB)



En formato comic...

Google

Web
 www.adictosaltrabajo.com
 Buscar

Tutorial desarrollado por



Enrique Viñé Lerma

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero Técnico en Informática por la Universidad Politécnica de Madrid.

Puedes encontrarme en [Autentia](#)

Somos expertos en Java/J2EE

Catálogo de servicios de Autentia

[Descargar \(6,2 MB\)](#)

[Descargar en versión comic \(17 MB\)](#)

AdictosAlTrabajo.com es el Web de difusión de conocimiento de Autentia.



[Catálogo de cursos](#)

Descargar este documento en formato PDF: [utilizaciondeguposenspringsecurity.pdf](#)

Últimos tutoriales

2008-12-16
[Utilización de grupos en Spring Security](#)

2008-12-17
[URLs amigables con UriRewriteFilter](#)

2008-12-10
[Modelado BPMN con Bizagi Modeler](#)

2008-12-10
[Tramites administrativos tras el nacimiento de un hijo](#)

2008-12-09
[Integración de Spring con el envío de emails: técnicas avanzadas \(II\)](#)

2008-12-09
[Reorganización estratégica](#)

2008-12-05
[Activación de los Dispositivos de Entrada en X.Org 1.5.3.](#)

2008-12-05
[Integración de Spring con el envío de emails: técnicas avanzadas \(I\)](#)

2008-12-01
[Weblets y como servir recursos que están en el CLASSPATH](#)

2008-12-03
[Edición de la Wikipedia y subida de Imágenes](#)

Fecha de creación del tutorial: 2008-12-16

Utilización de grupos en Spring Security

0. Índice de contenidos

- [1. Introducción](#)
- [2. Herramientas utilizadas](#)
- [3. Creando una aplicación de prueba](#)
- [4. Creando una base de datos para gestionar los usuarios, grupos y roles](#)
- [5. Configurando Spring para controlar la seguridad de nuestra aplicación](#)
- [6. Añadiendo usuarios, grupos y roles](#)
- [7. Conclusiones](#)

1. Introducción

En este primer tutorial que realizo en Autentia vamos a ver, con un sencillo ejemplo, como se configura Spring Security para controlar el acceso a los recursos de una aplicación, por medio de la asignación de roles a grupos de usuarios, los cuales se podrán modificar desde una base de datos.

Spring Security es un subproyecto del [framework Spring](#), que permite gestionar completamente la seguridad de nuestras aplicaciones Java, y cuyas ventajas principales son las siguientes:

- Es capaz de gestionar seguridad en varios niveles: URLs que se solicitan al servidor, acceso a métodos y clases Java, y acceso a instancias concretas de las clases.
- Permite separar la lógica de nuestras aplicaciones del control de la seguridad, utilizando filtros para las peticiones al servidor de aplicaciones o aspectos para la seguridad en clases y métodos.
- La configuración de la seguridad es portable de un servidor a otro, ya que se encuentra dentro del WAR o el EAR de nuestras aplicaciones.
- Soporta muchos modelos de identificación de los usuarios (HTTP BASIC, HTTP Digest, basada en formulario, LDAP, OpenID, Security Constraint, Ejemplos JSP, Java Basico)

Anuncios Google [Spring Framework](#) [Servlet Doget Dopost](#) [Security Constraint](#) [Ejemplos JSP](#) [Java Basico](#)

JAAS y muchos más). Además podemos ampliar estos mecanismos implementando nuestras propias clases que extiendan el modelo de Spring Security.

2. Herramientas utilizadas

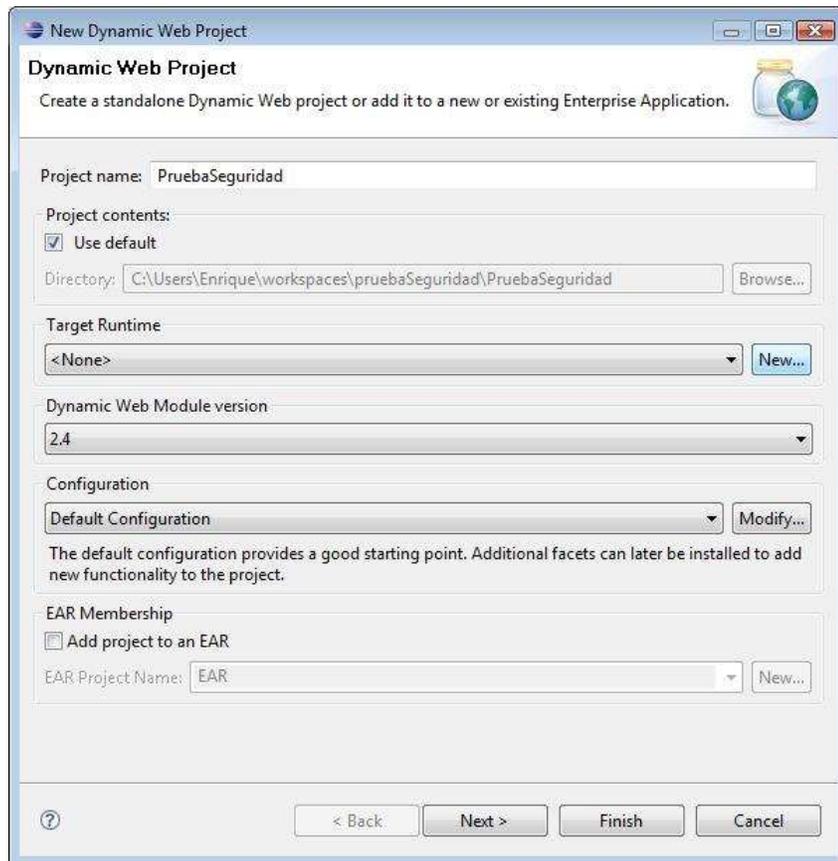
Este tutorial ha sido desarrollado en Windows Vista, y para su elaboración se han utilizado las siguientes herramientas:

- Eclipse Ganymede 3.4.1.
- MySql 5.1.30
- Connector/J de MySql
- Apache Tomcat 6.0.18

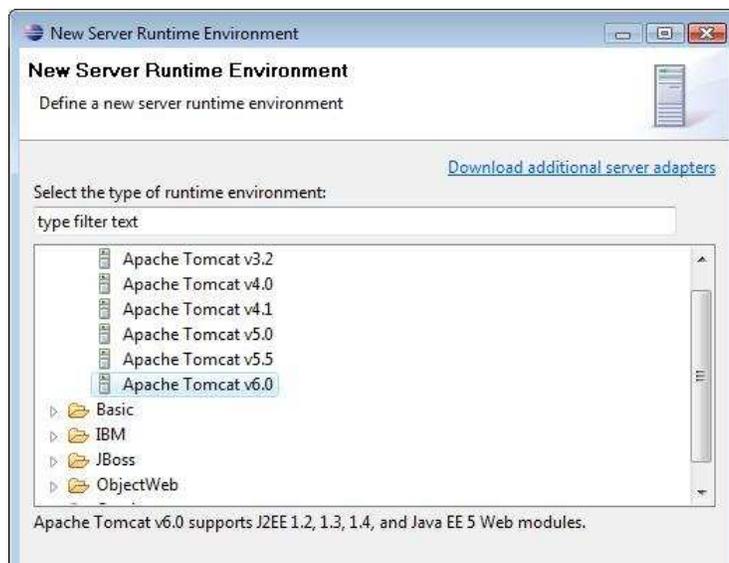
3. Creando una aplicación de prueba

Creamos un nuevo proyecto web dinámico. Podemos darle el nombre que queramos, en mi caso lo he llamado "PruebaSeguridad".

Vamos a seleccionar el Target Runtime pulsando sobre el botón new.



Seleccionamos como servidor Apache Tomcat 6.



Últimas ofertas de empleo

2008-11-27
Comercial - Ventas -
ALICANTE.

2008-10-30
Comercial - Ventas -
BARCELONA.

2008-10-30
T. Información - Analista /
Programador - BARCELONA.

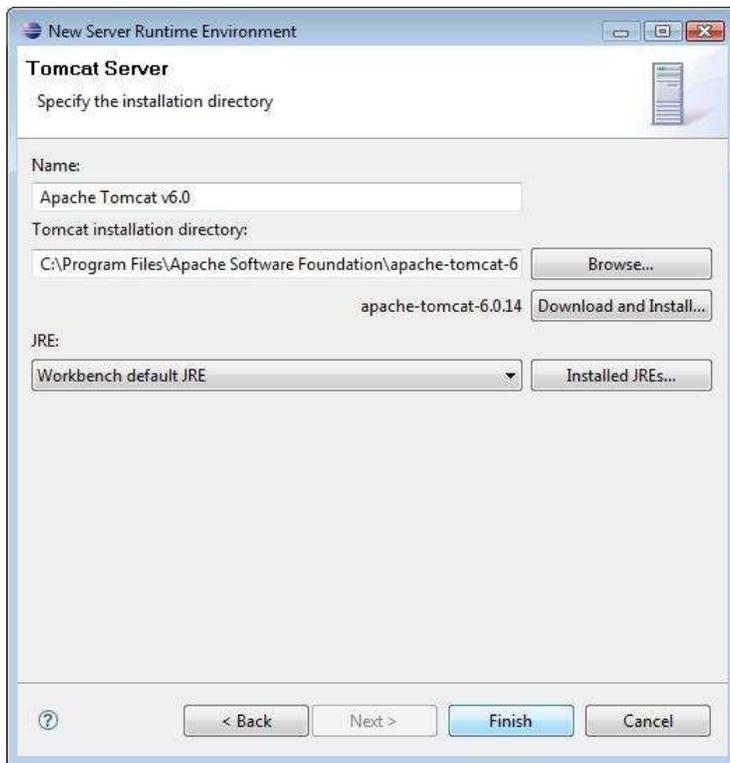
2008-10-27
T. Información - Analista /
Programador - CIUDAD REAL.

2008-10-03
Marketing - Experto en
Marketing - MADRID.

Anuncios Google



Seleccionamos la ruta donde tenemos instalado el tomcat, en mi caso "C:\Program Files\Apache Software Foundation\apache-tomcat-6.0.18", y pulsamos finalizar.



Vamos a crear tres sencillas páginas jsp de prueba dentro de la carpeta WebContent.

En la primera, "index.jsp", mostraremos un menú con dos opciones. Cada opción redirigirá a una de las otras dos páginas, a las cuales restringiremos el acceso más adelante por medio de Spring Security.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Inicio</title>
</head>
<body>

    <h1>Indice</h1>

    <p><a href="villancicos.jsp">Listado de villancicos</a></p>
    <p><a href="administracion.jsp">Administracion</a></p>
</body>
</html>
```

El contenido de las otras dos páginas podría ser cualquiera. En "villancicos.jsp", ya que se acerca la Navidad, mostraremos una lista con algunos villancicos populares, un enlace para poder volver al índice y otro para salir (y poder volver a identificarnos en la aplicación).

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Listado de villancicos</title>
</head>
<body>

    <h1>Listado de villancicos</h1>
    <table>
        <tr>
            <th>Titulo</th>
            <th>Popularidad</th>
        </tr>
        <tr>
            <td>Los peces en el río</td>
            <td>90</td>
        </tr>
    </table>
```

```

        <tr>
            <td>Campana sobre campana</td>
            <td>95</td>
        </tr>
        <tr>
            <td>La marimorena</td>
            <td>97</td>
        </tr>
    </table>

    <p><a href="index.jsp">Volver al indice</a></p>
    <p><a href="j_spring_security_logout">Salir</a></p>
</body>
</html>

```

La url "j_spring_security_logout" será capturada por el interceptor de Spring Security una vez que lo hayamos configurado, y permitirá hacer logout al usuario, de forma que si intenta acceder de nuevo a un recurso protegido, se le volverán a solicitar los datos de identificación.

En la página de administración, mostraremos un mensaje, un enlace para volver al índice y otro para salir.

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Administracion</title>
</head>
<body>

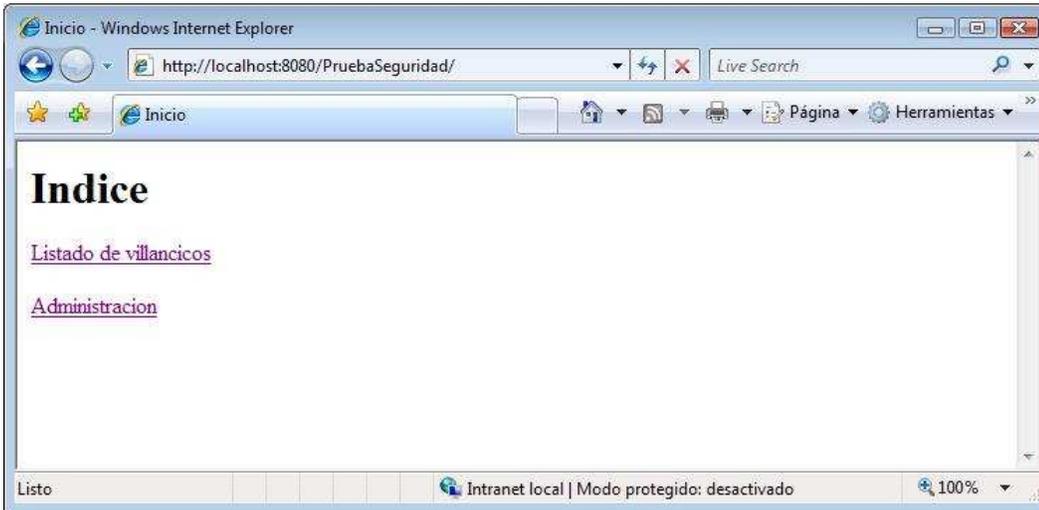
    <h1>Estás en la página de administración!</h1>

    <p><a href="index.jsp">Volver al indice</a></p>
    <p><a href="j_spring_security_logout">Salir</a></p>

</body>
</html>

```

Por último, para comprobar que todo funciona correctamente, ejecutamos nuestra pequeña aplicación en el Tomcat y desde un navegador probamos su funcionamiento.



4. Creando una base de datos para gestionar los usuarios, grupos y roles

Creamos un nuevo esquema de base de datos, utilizando una consola del sistema. Debemos utilizar un usuario con permiso para crear nuevos esquemas en la base de datos.

Para crear el esquema utilizaremos la sentencia: "create schema seguridad;"



```

C:\Program Files\MySQL\MySQL Server 5.1\bin>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 56
Server version: 5.1.30-community MySQL Community Server <GPL>

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create schema seguridad;
Query OK, 1 row affected (0.00 sec)

mysql> exit
Bye
C:\Program Files\MySQL\MySQL Server 5.1\bin>

```

Para crear las tablas hemos utilizado los siguientes scripts.

```

create table users(
  username varchar(50) not null primary key,
  password varchar(50) not null,
  enabled boolean not null);

create table authorities (
  username varchar(50) not null,
  authority varchar(50) not null,
  constraint fk_authorities_users foreign key(username) references users(username));

create unique index ix_auth_username on authorities (username,authority);

create table groups (
  id bigint auto_increment primary key,
  group_name varchar(50) not null);

create table group_authorities (
  group_id bigint not null,
  authority varchar(50) not null,
  constraint fk_group_authorities_group foreign key(group_id) references groups(id));

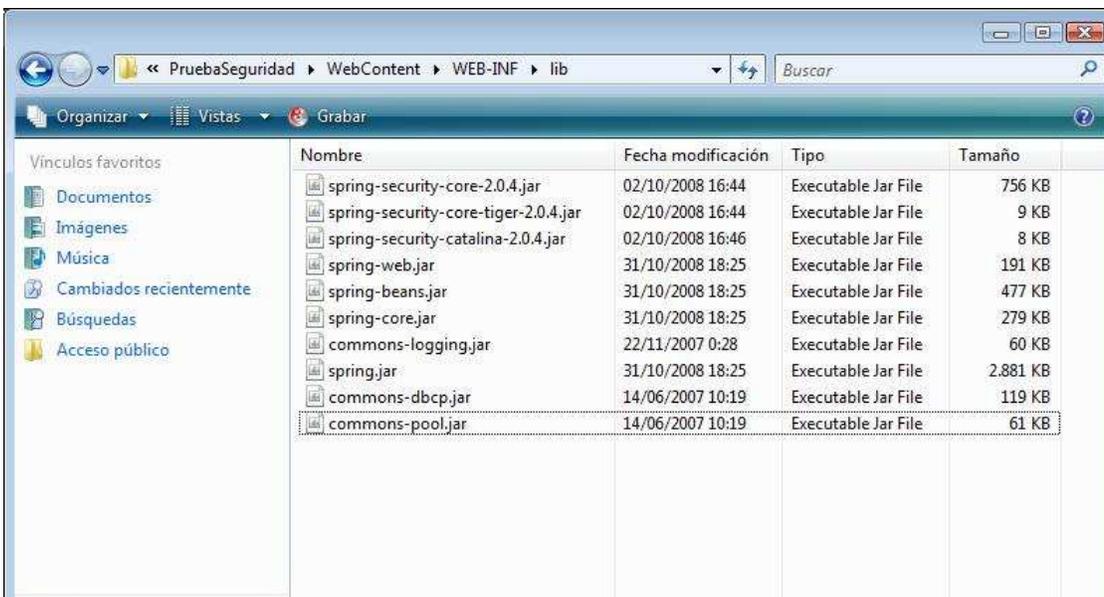
create table group_members (
  id bigint auto_increment primary key,
  username varchar(50) not null,
  group_id bigint not null,
  constraint fk_group_members_group foreign key(group_id) references groups(id));

```

NOTA: para que el servidor pueda encontrar el driver de MySql habrá que añadir el conector ("mysql-connector-java.jar") a la carpeta de librerías del Tomcat ("%TOMCAT_DIR%\lib").

5. Configurando Spring para controlar la seguridad de nuestra aplicación

Antes de nada, deberemos añadir dentro de "WebContent/lib" al menos las siguientes librerías, que se incluyen con Spring y Spring Security:



Nombre	Fecha modificación	Tipo	Tamaño
spring-security-core-2.0.4.jar	02/10/2008 16:44	Executable Jar File	756 KB
spring-security-core-tiger-2.0.4.jar	02/10/2008 16:44	Executable Jar File	9 KB
spring-security-catalina-2.0.4.jar	02/10/2008 16:46	Executable Jar File	8 KB
spring-web.jar	31/10/2008 18:25	Executable Jar File	191 KB
spring-beans.jar	31/10/2008 18:25	Executable Jar File	477 KB
spring-core.jar	31/10/2008 18:25	Executable Jar File	279 KB
commons-logging.jar	22/11/2007 0:28	Executable Jar File	60 KB
spring.jar	31/10/2008 18:25	Executable Jar File	2.881 KB
commons-dbc.jar	14/06/2007 10:19	Executable Jar File	119 KB
commons-pool.jar	14/06/2007 10:19	Executable Jar File	61 KB



Creamos el fichero de configuración para Spring Security, dentro de la carpeta WEB-INF, al que llamaremos "applicationContext-security.xml". El esqueleto básico será el siguiente:

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

Configuramos los interceptores de Spring Security, y establecemos los permisos adecuados para nuestras páginas:

- La página de índice no tendrá restringido el acceso.
- La página de administración tendrá el acceso restringido a los usuarios que tengan el rol "ROLE_ADMIN".
- El resto de páginas estarán restringidas a usuarios con el rol "ROLE_USER".

```
<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>
```

Spring Security aplicará la lista de interceptores en el orden en que aparece. Cuando encuentre un patrón adecuado, aplicará los permisos correspondientes y no continuará leyendo el resto de interceptores. Por tanto es importante colocar primero los filtros más específicos y después los más generales.

Además indicamos que nos muestre una página de login y el tipo de autenticación que vamos a utilizar, así como la página a la que se debe redirigir al hacer logout.

A continuación configuramos el AuthenticationProvider, que utilizará como UserService una de las implementaciones que vienen con Spring Security: JdbcDaoImpl. Esta implementación nos permite configurar los permisos de los usuarios en una base de datos a la que se accederá por medio de un DataSource.

```
<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:bean>
```

La propiedad "enableGroups" permite activar el uso de grupos para controlar la seguridad. Es importante cambiar el valor de esta propiedad si queremos utilizar grupos de seguridad, ya que por defecto vale false.

Añadimos el DataSource, indicando el usuario, contraseña y la URL a nuestro esquema de 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/seguridad?autoReconnect=true"
  p:username="root" p:password="root"/>
```

Por último debemos editar la configuración del fichero "web.xml" y añadir las siguientes líneas.

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>/WEB-INF/applicationContext-security.xml</param-value>
</context-param>
<listener>
  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
<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>
```

El parámetro de contexto "contextConfigLocation" indica la ruta o rutas de los ficheros de configuración de Spring. En nuestro caso indicamos la ruta del fichero de configuración creado anteriormente.

El ContextLoaderListener se encargará de cargar los ficheros de configuración de Spring indicados en "contextConfigLocation".

El filtro de seguridad es el encargado de manejar las peticiones que se hagan a todas las URLs para permitir o no el acceso de acuerdo a los permisos que hayamos establecido en el fichero de configuración.

Ahora podemos volver a arrancar nuestro servidor para probar los cambios. Al intentar acceder al listado de villancicos o a la página de administración se nos redirige a una página de login. Todavía no podremos acceder a estas páginas, ya que no hemos creado ningún usuario con los roles adecuados.



6. Añadiendo usuarios, grupos y roles

Ahora que ya está configurada nuestra aplicación de prueba, solamente nos queda añadir los usuarios, grupos y roles a la base de datos.

Añadimos los siguientes usuarios a la tabla USERS:

```
username  password  enabled
-----
'enrique' 'enrique' 1
'juan'    'juan'    1
'lucas'   'lucas'   0
'pepe'    'pepe'    1
'rosa'    'rosa'    1
```

Añadimos los siguientes grupos a la tabla GROUPS:

```
id  group_name
---
1   'usuarios'
2   'administradores'
3   'invitados'
```

A continuación podemos dar roles a los grupos, añadiendo los siguientes registros en la tabla GROUP_AUTHORITIES, que relaciona roles con grupos. En nuestro ejemplo hemos añadido un único rol a cada grupo, pero es posible asignar n roles a cada uno. El grupo 'invitados' no tiene ningún rol asignado, por lo que los usuarios que pertenezcan a él no tendrán permisos definidos y no podrán acceder a los recursos protegidos.

```
group_id  authority
-----
1         'ROLE_USER'
2         'ROLE_ADMIN'
```

Por último, añadimos los usuarios a los grupos, desde la tabla GROUP_MEMBERS, que relaciona grupos con usuarios. Cada usuario podrá pertenecer a varios grupos y sus roles efectivos serán la unión de los roles de cada grupo a los que pertenezca.

```
id  username  group_id
---
2   'lucas'   2
3   'juan'    1
4   'rosa'    3
5   'enrique' 2
6   'pepe'    1
8   'enrique' 1
```

De esta forma los usuarios "juan" y "pepe" pertenecerán al grupo "usuarios", mientras que "lucas" pertenecerá al de "administradores". "rosa" pertenece al grupo invitados, que no tiene asignado ningún permiso, mientras que "enrique" pertenece tanto al grupo de administradores como al de usuarios.

Efectivamente, no hemos utilizado la tabla AUTHORITIES para nada. Esta tabla permite asignar roles directamente a los usuarios, pero en nuestro caso hemos preferido realizar esta asignación a través de los grupos. No obstante, todavía podríamos asignar permisos directamente a los usuarios si utilizamos esta tabla, de forma que el usuario tendría la suma de sus permisos más los que tengan los grupos a los que pertenece. Si queremos deshabilitar la asignación de permisos directamente a los usuarios (es decir que no se tenga en cuenta la tabla AUTHORITIES) tendremos que asignar el valor false a la propiedad "enableAuthorities" de nuestro 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>
```

La propiedad "enableAuthorities" toma el valor true por defecto, al contrario que "enableGroups". Es importante dejar al menos una de las dos propiedades (enableGroups o enableAuthorities) a true, ya que de lo contrario obtendremos un error de configuración de Spring Security.

Podemos intentar entrar en la aplicación con los distintos usuarios para ver si obtenemos el comportamiento esperado.

Vemos que cualquiera puede acceder a la página de índice, incluso aunque no se haya identificado, pero cuando pulsamos en uno de los enlaces se nos muestra la página de login.

Cuando probamos a acceder con el usuario "lucas" se nos muestra un mensaje indicando que el usuario está deshabilitado (está en inglés, ya que es un formulario de login que implementado por defecto Spring Security).





Cuando probamos con "rosa" ocurre algo parecido, pero esta vez el mensaje es diferente, ya que el usuario no tiene los permisos adecuados.



Por último podemos comprobar que "juan" y "pepe" pueden acceder solamente al listado de villancicos, pero cuando intentan acceder a la página de administración se muestra el siguiente error, ya que no tienen permiso para acceder a la página.



El único usuario al que se permite acceder a la página de administración es "enrique", ya que lo hemos añadido al grupo de administradores.

7. Conclusiones

A la vista de los resultados de este tutorial podemos sacar las siguientes conclusiones:

- Spring Security nos ofrece servicios de identificación y acceso a los recursos de nuestras aplicaciones de una forma potente y sencilla, sin tener que escribir ni una sola línea de código.
- Podemos asignar los permisos directamente a los usuarios o a grupos de usuarios, o incluso mezclar ambas formas.

- Podemos añadir fácilmente seguridad en nuestras aplicaciones existentes o modificar la seguridad de una aplicación rápidamente, ya que Spring Security separa claramente la seguridad de una aplicación de su lógica.

Y eso es todo. Desde autentia esperamos que este tutorial os haya sido de utilidad y os ayude a gestionar la seguridad de vuestras aplicaciones mediante Spring Security.

- Puedes opinar sobre este tutorial [haciendo clic aquí](#).
- Puedes firmar en nuestro libro de visitas [haciendo clic aquí](#).
- Puedes asociarte al grupo AdictosAlTrabajo en XING [haciendo clic aquí](#).
- Añadir a favoritos Technorati. 



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com



Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de suscripción a novedades:

E-mail

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
Seguridad en Tomcat	Os mostramos como proteger de un modo básico el acceso a recursos dentro de vuestro servidor de componentes Tomcat	2003-07-07	39145	pdf
Integración de Spring con el envío de emails	Nuestro compañero Jose, continuando con la saga de tutoriales de Spring, nos enseña en ésta ocasión la integración con un servicio de correo electrónico	2008-11-24	687	pdf
Planificación de tareas con Spring	Spring nos proporciona varias formas de planificar las tareas a través de Quartz, y en este tutorial Juan nos enseña un ejemplo práctico	2008-10-10	1273	pdf
Crear un logger utilizado a través de aspectos con Spring AOP.	En este tutorial os enseñamos cómo implementar un logger utilizado a través de aspectos con Spring AOP.	2008-02-22	2349	pdf
Creación de una aplicación con Spring e Hibernate desde 0	Este tutorial vamos a explicar paso a paso cómo crear una pequeña aplicación usando Spring e Hibernate con anotaciones partiendo desde 0	2008-02-15	8564	pdf
SpringIDE, plugin de Spring para Eclipse	En adictosaltrabajo os hemos ido presentando diversos plugins para Eclipse. Esta vez le toca el turno a SpringIDE, un plugin que os ayudará a desarrollar aplicaciones que utilicen Spring.	2008-01-19	5249	pdf
Spring + Hibernate + Anotaciones = Desarrollo Rápido en Java	Alejandro Pérez nos enseña lo fácil y rápido que es desarrollar en Java usando Spring e Hibernate, y usando anotaciones	2008-05-14	8292	pdf
Spring: definición dinámica de Beans	Este tutorial habla sobre la modificación dinámica de los beans del contexto para simplificar la configuración de Spring	2007-05-09	5365	pdf
Creación de una aplicación web con SpringMVC desde 0	Este tutorial te resultará muy útil para aprender a usar el patrón modelo-vista-controlador (MVC) con Spring a nuestros desarrollos web	2008-05-05	4306	pdf
Comparativa entre EJB3 y Spring	En este tutorial os mostramos una comparativa entre EJB3 y Spring esperando que os ayude a decidir qué tecnología utilizar.	2007-10-17	5399	pdf

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.