Aplicación Web

Este artículo se concentra en características del servidor Apache Geronimo, relacionadas a una aplicación web. La aplicación ejemplo es un sistema básico que reporta tiempos, usando Servlets, JSPs y seguridad declarativa J2EE. Además de las características previas, la aplicación usa la base de datos Derby embebida en Geronimo para guardar información del usuario del sistema. Aunque esta aplicación usa base de datos para guardar información del usuario, su uso primordial es por cuestiones de configuración. Para información detallada en el uso de JDBC en Geronimo, consulta el artículo Aplicación ejemplo de acceso simple a base de datos.

Después de leer este artículo deberías ser capaz de configurar al servidor de aplicaciones Geronimo para aplicaciones web con características de seguridad declarativa.

Este artículo se organiza en las secciones siguientes.

Aplicaciones Web en Geronimo

Apache Geronimo incluye un contenedor de aplicaciones Web que soporta aplicaciones Web J2EE. El contenedor Web por si mismo soporta configuraciones básicas como puertos de red y opciones SSL, y cada aplicación Web también podría incluir información de configuración Geronimo-específica. Las aplicaciones Web participan en la infraestructura de seguridad Geronimo, tal que autenticándose a una aplicación Web permite acceso a EJBs seguros así como a Conectores.

Actualmente, Apache Geronimo soporta dos contenedores Web: Jetty and Tomcat.

Jetty

Jetty es un Servidor 100% Java HTTP y Contenedor de Servlet. Esto significa que no necesitas configurar y correr un servidor Web por separado para usar servlets y JSPs con el fin de generar contenido dinámico: Jetty es un servidor Web completamente funcional para contenido estático y dinámico.

A diferencia de soluciones separadas servidor/contenedor, el servidor Web Jetty y la aplicación Web se ejecutan en el mismo proceso sin cargas de interconexión ni complicaciones. Además, como es un componente java puro, Jetty puede ser fácilmente incluido en tu aplicación para demostración, distribución ó activación. Jetty esta disponible para toda plataforma con soporte Java.
http://jetty.mortbay.org/jetty/index.html

Tomcat

Apache Tomcat es el contenedor de servlet que es usado en la Referencia Oficial de Implementación para tecnologías de Java Servlet y JavaServer Pages.
http://tomcat.apache.org/

Visión General de la Aplicación

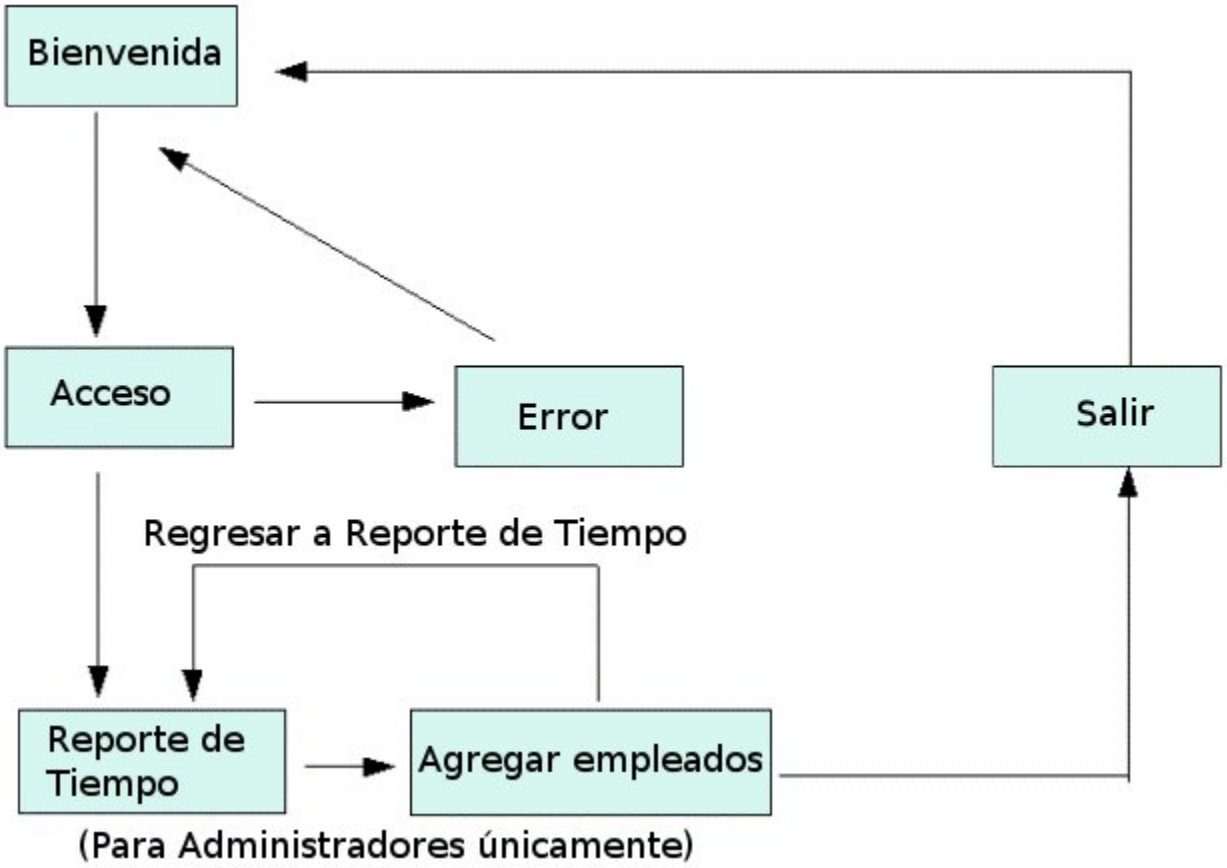
La aplicación de Reporte de Tiempo ayuda a reportar el tiempo de trabajo de distintos proyectos. Aunque no es una aplicación de reporte total de tiempo, cubre la mayoría de características, de despliegue y seguridad, relacionadas con aplicaciones Web en Apache Geronimo.

Esta aplicación ejemplo te permite dos tipos de grupos de usuario para reportar el tiempo de sus actividades al sistema, llamados administradores y empleados. Ambos tipos de usuarios deben proporcionar sus credenciales antes de reportar el tiempo de actividades. Administradores son también los super usuarios del sistema, tal que también pueden agregar empleados al sistema.

La aplicación de Reporte de Tiempo tiene la siguiente lista de páginas.

- Bienvenida
- Acceso
- Reporte de Tiempo
- Agregar Empleados
- Salida

La siguiente figura ilustra, de forma general, el flujo de la aplicación:



Por defecto, la aplicación ejemplo se redirige a la páginde de Bienvenida con una liga a la funcionalidad del Reporte de Tiempo. Los usuarios pueden tener acceso a la página de Reporte de Tiempo al proporcionar nombre de usuario y contraseña válidos en la página de Acceso. Si las credenciales del usuario pertenecen a un rol administrador, la página de Reporte de Tiempo también desplegará una liga adicional para la funcionalidad de Agregar Empleados.

Contenido de la Aplicación

A continuación se muestra la jerarquía principal de folders en la aplicación del Reporte de Tiempo. Despliega tanto a los JSPs como a los archivos de configuración usados en la aplicación.

```
|- employee
     |- index.jsp
|- login
     |- login.jsp
     |- login_error.jsp
     |- logout.jsp
|- manager
     |- index.jsp
|- WEB_INF
     |- geronimo-web.xml
     |- web.xml
|- index.jsp
```

Además de las configuraciones de JSPs, se requieren otros dos servlets para completar la lógica de negocios de la aplicación.

- AddTimeRecordServlet - Lee los datos de entrada de la página de Reporte de Tiempo
- AddEmployeeServlet - captura la información de entrada de la página de Agregar Empleado

La configuración de Seguridad en la aplicación de Reporte de Tiempo es controlada por los archivos **geronimo-web.xml** y **web.xml**. **geronimo-web.xml** se usa para definir roles de usuario de la aplicación con **TimeReportRealm**.

geronimo-web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
        xmlns="http://geronimo.apache.org/xml/ns/j2ee/web-1.1">

        <environment>
                <moduleId>
                        <artifactId>TimeReportApp</artifactId>
                </moduleId>
        </environment>

        <context-root>/timereport</context-root>

        <security-realm-name>TimeReportRealm</security-realm-name>

        <security>
                <default-principal realm-name="TimeReportRealm">
                        <principal name="anonymous"
                                   class="org.apache.geronimo.security.realm.providers.GeronimoUserPrincipal"
                                   />
                </default-principal>
                <role-mappings>
                        <role role-name="employee">
                                <realm realm-name="TimeReportRealm">
                                        <principal name="EmployeeGroup"
                                           class="org.apache.geronimo.security.realm.providers.
GeronimoGroupPrincipal"
                                           />
                                </realm>
                                <realm realm-name="TimeReportRealm">
                                        <principal name="ManagerGroup"
                                           class="org.apache.geronimo.security.realm.providers.
GeronimoGroupPrincipal"
                                        />
                                </realm>
                        </role>
                        <role role-name="manager">
                                <realm realm-name="TimeReportRealm">
                                        <principal name="ManagerGroup"
                                           class="org.apache.geronimo.security.realm.providers.
GeronimoGroupPrincipal"
                                        />
                                </realm>
                        </role>
                </role-mappings>
    </security>

</web-app>
```

web.xml mapeará los roles de usuario definidos a los recursos en la aplicación web. También define una configuración de acceso a la aplicación.

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
         http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
         version="2.4">

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

        <security-constraint>
                <web-resource-collection>
                        <web-resource-name>employee</web-resource-name>
                        <url-pattern>/employee/*</url-pattern>
                </web-resource-collection>
                <auth-constraint>
                        <role-name>employee</role-name>
                </auth-constraint>
        </security-constraint>

        <security-constraint>
                <web-resource-collection>
                        <web-resource-name>manager</web-resource-name>
                        <url-pattern>/manager/*</url-pattern>
                </web-resource-collection>
                <auth-constraint>
                        <role-name>manager</role-name>
                </auth-constraint>
        </security-constraint>

        <login-config>
                <auth-method>FORM</auth-method>
                <realm-name>TimeReportRealm</realm-name>
                <form-login-config>
                        <form-login-page>/login/login.jsp</form-login-page>
                        <form-error-page>/login/login_error.jsp</form-error-page>
                </form-login-config>
        </login-config>

        <security-role>
                <role-name>employee</role-name>
            </security-role>
        <security-role>
                <role-name>manager</role-name>
    </security-role>

    <servlet>
            <display-name>AddTimeRecordServlet</display-name>
            <servlet-name>AddTimeRecordServlet</servlet-name>
            <servlet-class>org.timereport.web.employee.AddTimeRecordServlet</servlet-class>
          </servlet>
          <servlet>
            <display-name>AddEmployeeServlet</display-name>
            <servlet-name>AddEmployeeServlet</servlet-name>
            <servlet-class>org.timereport.web.manager.AddEmployeeServlet</servlet-class>
          </servlet>

          <servlet-mapping>
            <servlet-name>AddTimeRecordServlet</servlet-name>
            <url-pattern>/employee/add_timerecord</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
            <servlet-name>AddEmployeeServlet</servlet-name>
            <url-pattern>/manager/add_employee</url-pattern>
    </servlet-mapping>

</web-app>
```

Para restringir el acceso a la funcionalidad de Agregar Empleado en la página de Reporte de Tiempo, autenticación programática ha sido usada como se indica a continuación.

employee/index.jsp

```
...
<BR>
<%if(request.isUserInRole("manager")){%>
<A href="../manager/">Add Employees</A>
<BR>
...
```

Herramientas usadas

Las herramientas usadas para el desarrollo y construcción de un ejemplo de aplicación de Reporte de Tiempo, son:

Eclipse

El IDE Eclipse fue empleado para el desarrollo de la aplicación ejemplo. Es una herramienta de desarrollo muy poderosa y popular. También tiene plug-ins de integración a Geronimo. Eclipse puede descargarse de la siguiente URL:
http://www.eclipse.org

Apache Ant

Ant es una herramienta 100% Java de construcción. Se usa para la construcción de archivos war para la aplicación de Reporte de Tiempo. Ant puede descargarse de la siguiente URL:
http://ant.apache.org

Regresar a la sección superior

----------> FALTA TRADUCCION

Configuring, Building and Deploying the Sample Application

Download the Time Reporting application from the following link:
Time Report

After extracting the zip file, the <time_report> directory is created.

Configuring

Since Time Reporting application is going to use J2EE declarative security, user needs to create a database to hold information and deploy security realm.

Create Database to hold User Information

After starting Apache Geronimo server, log in to the console and follow the given steps to create the **TimeReportDB** to hold user information for the application.

TimeReportDB.sql

```
CREATE TABLE users(
        userid VARCHAR(15) PRIMARY KEY,
        password VARCHAR(15),
        name VARCHAR(40)
);

CREATE TABLE usergroups(
        userid VARCHAR(15),
        groupname VARCHAR(20),
        PRIMARY KEY (userid, groupname)
);

INSERT INTO users VALUES('emp1', 'pass1', 'Employee 1');
INSERT INTO users VALUES('emp2', 'pass2', 'Employee 2');
INSERT INTO users VALUES('mgm1', 'pass3', 'Manager 1');
INSERT INTO users VALUES('mgm2', 'pass4', 'Manager 2');

INSERT INTO usergroups VALUES('emp1', 'EmployeeGroup');
INSERT INTO usergroups VALUES('emp2', 'EmployeeGroup');
INSERT INTO usergroups VALUES('mgm1', 'ManagerGroup');
INSERT INTO usergroups VALUES('mgm2', 'ManagerGroup');
```

1. Select **DB Manager** link from the **Console Navigation** in the left.
2. Give the database name as **TimeReportDB** in the **Create DB** field and click **Create** button.
3. Select TimeReportDB to the **Use DB** field.
4. Open **TimeReportDB.sql** in the **time_report/config** directory.
5. Paste the content **TimeReportDB.sql** to the **SQL Commands** text area and press **Run SQL** button.

Configure Security Realm

As same as in the creating database, follow the given steps to deploy the security relam of the Time Reporting application.

TimeReportRealm.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="http://geronimo.apache.org/xml/ns/deployment-1.1">
    <environment>
        <moduleId>
            <groupId>console</groupId>
            <artifactId>TimeReportRealm</artifactId>
            <version>1.0</version>
            <type>car</type>
        </moduleId>
        <dependencies>
            <dependency>
                <groupId>geronimo</groupId>
                <artifactId>j2ee-security</artifactId>
                <type>car</type>
            </dependency>
            <dependency>
                <groupId>org.apache.derby</groupId>
                <artifactId>derby</artifactId>
                <version>10.1.1.0</version>
                <type>jar</type>
            </dependency>
        </dependencies>
    </environment>
    <gbean name="TimeReportRealm" class="org.apache.geronimo.security.realm.GenericSecurityRealm">
        <attribute name="realmName">TimeReportRealm</attribute>
        <reference name="ServerInfo">
            <name>ServerInfo</name>
        </reference>
        <reference name="LoginService">
            <name>JaasLoginService</name>
        </reference>
        <xml-reference name="LoginModuleConfiguration">
            <log:login-config xmlns:log="http://geronimo.apache.org/xml/ns/loginconfig-1.1">
                <log:login-module control-flag="REQUIRED" server-side="true" wrap-principals="false">
                    <log:login-domain-name>TimeReportRealm</log:login-domain-name>
                    <log:login-module-class>org.apache.geronimo.security.realm.providers.SQLLoginModule</log:
login-module-class>
                    <log:option name="jdbcDriver">org.apache.derby.jdbc.EmbeddedDriver</log:option>
                    <log:option name="jdbcUser">app</log:option>
                    <log:option name="userSelect">select userid, password from users where userid=?</log:option>
                    <log:option name="groupSelect">select userid, groupname from usergroups where userid=?</log:
option>
                    <log:option name="jdbcURL">jdbc:derby:TimeReportDB</log:option>
                </log:login-module>
            </log:login-config>
        </xml-reference>
    </gbean>
</module>
```

1. Select **Deploy New** link from **Console Navigation**.
2. Load **time_report/config/TimeReportRealm.xml** to the **Plan** input box.
3. Press **Install** button deploy security realm to the application server.
   (Make sure **Start app after install** check box is in selected state before pressing install button.)

For the verification after the deployment, go to the **Security Realms** in Geronimo Console and see new security realm **TimeReportRealm** is there.

Building

Time Report application comes with an Ant script to help users to build from source code. Use a command prompt to travel to the **time_report** directory and just give **ant** command to build. It will create a **TimeReport.war** under the **releases** folder in the **time_report**. Now you are ready to deploy the Time Report application in the Geronimo Application server.

Deploying

Deploying the sample application is pretty straight forward, since we are using the Geronimo Console.

1. Travel **Deploy New** from the **Console Navigation**.
2. Load **TimeReport.war** from **time_report/releases** folder in to the **Archive** input box.
3. Press **Install** button to deploy application in the server.

Regresar a la sección superior

Testing of the Sample Application

To test the sample application open a browser and type http://localhost:8080/timereport. It will forward to the Welcome page of the application.

User can access Time Report page providing username as **emp1** and password with **pass1**. To login to the application as a Manager provide **mgm1** and **pass3** credentials.

Summary

This article has shown you how to deploy web application in to the Geronimo Application server with J2EE declarative security features. You followed step-by-step instructions to build, deploy and test the sample application.

Some highlights of the article are:-

- Apache Geronimo provides two different web containers namely Jetty and Tomcat.
- Create database to hold security data with built-in Derby.
- Define security roles in Geronimo Web applications.
- Deploy deployment plans and web archives using Geronimo Console.
  <---------- FALTA TRADUCCION