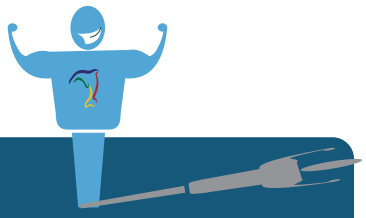
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

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

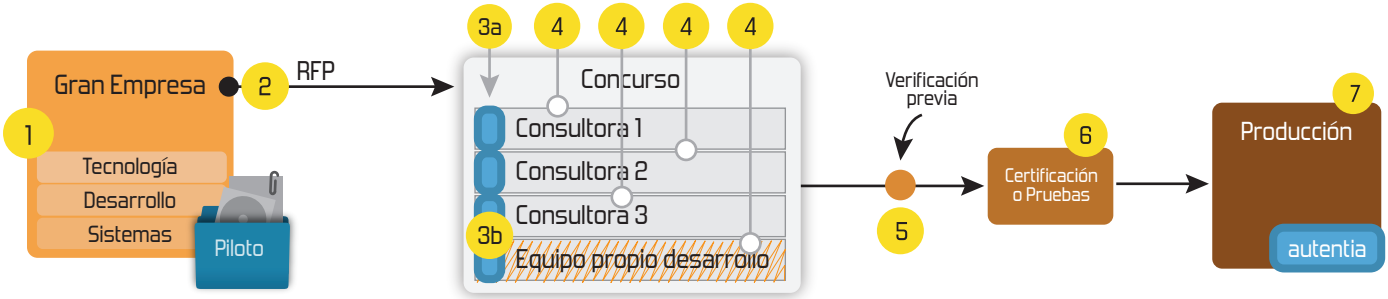
1. Desarrollo de componentes y proyectos a medida



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

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

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

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

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

1. **INTRODUCCIÓN**

   En este tutorial vamos a explicar como activar Single Sign On en Jboss 4.2.2 en dos aplicaciones web diferentes. Los pasos que realizaremos para llevar a cabo dicha funcionalidad serán:

   - Crearemos dos aplicaciones llamadas Web1 y Web2.
     - Configuraremos JAAS protegiendo los recursos necesarios bajo los roles adecuados
     - Indicaremos la forma de autentificación
   - Crearemos un dominio de seguridad de Jboss bajo el que podremos ambas aplicaciones.
   - Activaremos la válvula de tomcat necesaria para activar Single Sign On.

2. **CREACIÓN DE LAS APLICACIONES WEB.**

   He creado dos aplicaciones Web (usando Eclipse Europa con las Jboss Tools instaladas aunque esto no es necesario para el tutorial) con la siguiente configuración en el descriptor web.xml

   Descriptor de la Web1

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://java.sun.com/xml/ns/javaee"
         xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2
         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
         id="WebApp_ID"
         version="2.5">
  <display-name>Web1</display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
  <security-role>
    <role-name>Administrador</role-name>
  </security-role>
  <security-constraint>
    <web-resource-collection>
        <web-resource-name>todo</web-resource-name>
        <url-pattern>/pages/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <description>Sólo admin</description>
        <role-name>Administrador</role-name>
    </auth-constraint>
  </security-constraint>
  <login-config>
    <auth-method>FORM</auth-method>
    <realm-name>MI_REINO_1</realm-name>
    <form-login-config>
        <form-login-page>/login.jsp</form-login-page>
        <form-error-page>/errorValidacion.jsp</form-error-page>
    </form-login-config>
  </login-config>
</web-app>
```

   Descriptor de la Web2

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://java.sun.com/xml/ns/javaee"
         xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_
         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee h
         id="WebApp_ID"
         version="2.5">
  <display-name>Web2</display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
   <security-role>
    <role-name>Administrador</role-name>
  </security-role>
  <security-constraint>
  <web-resource-collection>
    <web-resource-name>todo</web-resource-name>
    <url-pattern>/pages/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <description>Sólo admin</description>
    <role-name>Administrador</role-name>
  </auth-constraint>
  </security-constraint>
  <login-config>
    <auth-method>FORM</auth-method>
    <realm-name>MI_REINO_2</realm-name>
    <form-login-config>
        <form-login-page>/login.jsp</form-login-page>
        <form-error-page>/errorValidacion.jsp</form-error-page>
    </form-login-config>
  </login-config>
</web-app>
```

   En ambos casos:

   - Las aplicaciones definen una página por defecto index.jsp
   - Ambas aplicaciones definen el rol de Administrador
   - Ambas aplicaciones deciden proteger todo lo que cuelga del directorio “pages” y dar permisos únicamente al rol Administrador
   - Ambas aplicaciones deciden configurar la autentificación por formulario, indicando la página de login y la página de error de validación.

   La página de inicio (index.jsp) en ambos casos:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%
    response.sendRedirect("pages/home.jsp");
%>
```

   La página de login (login.jsp). Es importante que los nombres de los cambos y el action del formulario sean como los indicados.

```
<%@ 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.c
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>LOGIN</title>
</head>
<body>
<div id="layout">
<form method="post" action="j_security_check" name="j_security_check">
<table id="loginTable">
    <tbody>
        <tr>
            <td class="loginLabel"><span>Usuario</span></td>
            <td><input type="text" name="j_username" /></td>
        </tr>
        <tr>
            <td class="loginLabel"><span>Contraseña</span></td>
            <td><input type="password" name="j_password" /></td>
        </tr>
        <tr>
            <td colspan="2"><input type="submit" value="Entrada al sistema" />
            </td>
        </tr>
```

```
    </tbody>
</table>
</form>
```

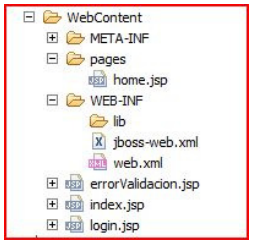
La página de error (errorValidacion.jsp).

```
<%@ 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://ww
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Error de validacion</title>
</head>
<body>
Has fallado.
</body>
</html>
```

La página Home (pages/home.jsp).

```
<%@ 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.
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
Home Web2
</body>
</html>
```

La estructura de las aplicaciones será:



3. **DOMINIO DE SEGURIDAD**

Indicaremos a JBoss que queremos poner a ambas Webs bajo el mismo dominio de seguridad:

Descriptor jboss-web.xml de Web1

```
<jboss-web>
    <security-domain>
        java:/jaas/MiAuthenticationPolicy
    </security-domain>
    <context-root>/Web1</context-root>
</jboss-web>
```

Descriptor jboss-web.xml de Web2

```
<jboss-web>
    <security-domain>
        java:/jaas/MiAuthenticationPolicy
    </security-domain>
    <context-root>/Web2</context-root>
</jboss-web>
```

A continuación debemos crear nuestro dominio de seguridad "MiAuthenticationPolicy". Para ello podéis guiaros por el siguiente tutorial.

4. **ACTIVAR SINGLE SIGN ON**

Esta es la parte más sencilla de realizar. Para activarlo accederemos al fichero de configuración del tomcat integrado con JBoss y activaremos la válvula apropiada. El fichero se encuentra en: JBOSS_HOME/server/tu_configuracion/deploy/jboss-web.deployer/server.xml
Editaremos el fichero y descomentaremos la válvula siguiente:

<Valve className="org.apache.catalina.authenticator.SingleSignOn" />

Si ahora reiniciais vuestro servidor y desplegáis ambas aplicaciones, podréeis comprobar que cuando os autentifiquéis en una, ya lo estaréis en la otra. Sin embargo, si hicieseis un logout de una (borrais la sesión) no os desconectaréis de la otra y viceversa (lo mismo pasa con el timeout de sesión)

Si lo que teneis es un cluster, deberíais desactivar esta válvula y activar:
<Valve className="org.jboss.web.tomcat.service.sso.ClusteredSingleSignOn" />