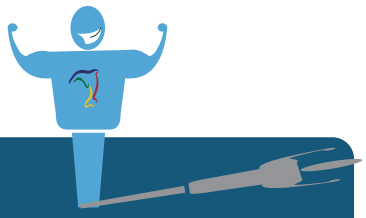
autentia

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...

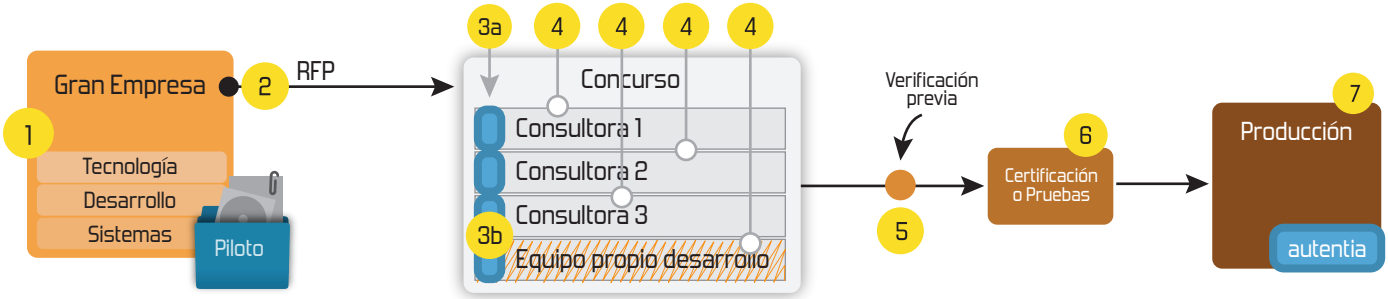
1. Desarrollo de componentes y proyectos a medida



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 conociemiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

adictos al trabajo Powered by autentia Feliz año 2008 Hosting Patrocinado por enREDados.com enREDados

Ver cursos que ofrece Autentia - - Descargar comics en PDF y alta resolución

Tutorial desarrollado por:

Francisco Javier Martínez Páez

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en Autentia

Somos expertos en Java/J2EE

Nuevo catálogo de servicios de Autentia

Descargar catálogo (6,2 MB)

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

autentia real business solutions

Catálogo de cursos

Descargar este documento en formato PDF IceJboss.pdf

Trabaje desde casa - Ingresos extra desde casa.
Tiempo completo o parcial. www.trabajaportucuenta.com

Anuncios Google

Fecha de creación del tutorial: 2008-01-17

Icefaces, JBoss, Maven2 y EJB3: Parte 1.

Voy a realizar una serie de tutoriales encadenados, que pretenden obtener un arquetipo de proyecto con las siguientes características:

- IDE: Eclipse 3.1.1
- Ciclo de vida del proyecto: Maven2. (usaré el plugin m2eclipse para gestionar maven2 y eclipse)

- Parte de la Vista / Controlador: usaré JSF RI (incluida en JBoss) y icefaces
- Parte del Modelo: EJB 3.0
- Servidor de aplicaciones: JBoss 4.2.2
- Base de datos: MySQL

En esta primera parte, nuestro objetivo será crear la estructura de proyectos en eclipse, crear los ficheros pom.xml de cada uno de ellos con las dependencias para empezar a trabajar.

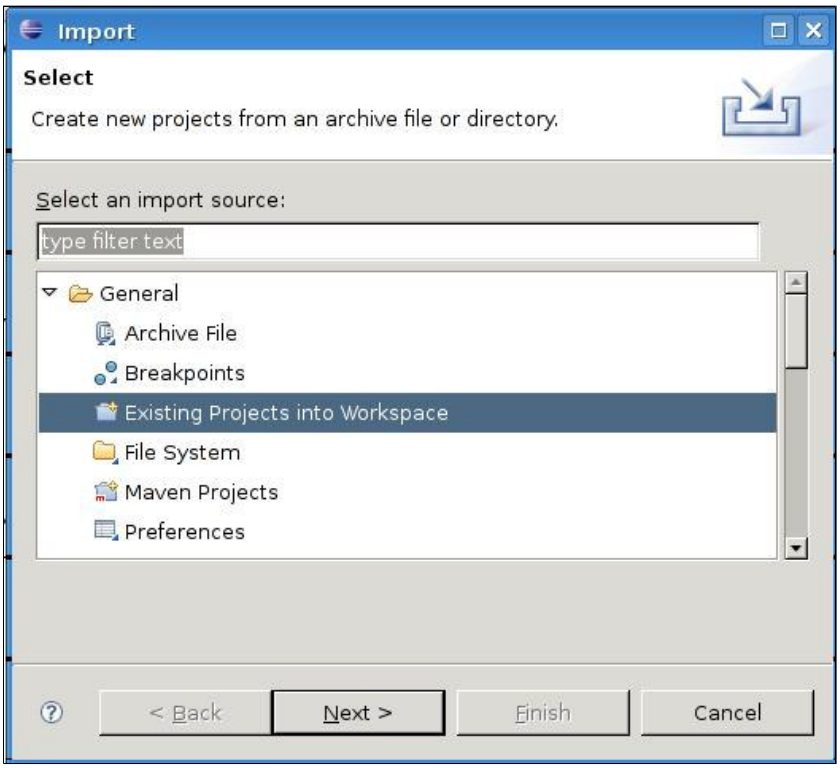
1. **Creando estructura de proyecto en eclipse:**

   Existen ya algunos tutoriales de Alejandro Pérez explicando como llevar a cabo esta tarea para que Eclipse y Maven2 puedan superar sus diferencias:

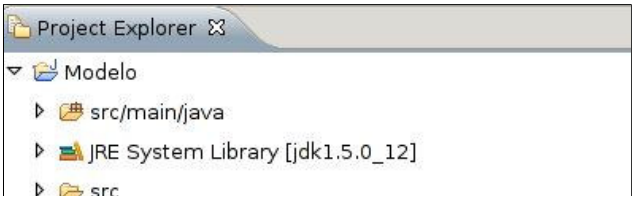
   http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=mavenRelease

   Básicamente, lo que vamos a hacer:

- Crearemos un proyecto de propósito general y lo denominaremos 'Padre'.
- Crearemos un proyecto EJB y lo denominaremos 'Modelo'
- Crearemos un proyecto web dinámico (Dynamic Web Project) y lo denominaremos 'Web'
- Borraremos los proyectos 'Modelo' y 'Web' (¡OJO!, NO borraremos su contenido.)
- Usando el explorador de archivos de tu Sistema Operativo favorito, moveremos ambos proyectos ('Web' y 'Modelo') dentro de 'Padre'
- Importaremos los proyectos 'Web' y 'Modelo' desde Eclipse: File â€" Import:



Nos habrá quedado una estructura como muestra la imagen:



2. **Creando los ficheros pom.xml**

Empezaremos por el proyecto Padre, que será el que englobe a todos los módulos. En la raíz del proyecto, crearemos un fichero pom.xml:

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

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.autentia.tutoriales.icefaces</groupId>

<artifactId>Padre</artifactId>

<version>1.0-SNAPSHOT</version>

<packaging>pom</packaging>

<name>Padre</name>

<description>

Este proyecto es el padre de todas los módulos.

Mantiene configuraciones comunes.

</description>

<modules>

<module>Web</module>

<module>Modelo</module>

</modules>

<repositories>

<repository>

<id>java-net</id>

<name>java-net</name>

<url>http://download.java.net/maven/2</url>

</repository>

<repository>

<id>java.net</id>

<url>http://download.java.net/maven/1</url>

<layout>legacy</layout>

</repository>

<repository>

<id>jboss</id>

<url>http://repository.jboss.com/maven2/</url>

</repository>
```

```
</repositories>

<dependencies>

<!-- Comunes a los módulos -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.14</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

<version>1.1</version>

</dependency>

</dependencies>

</project>
```

A continuación, crearemos el del proyecto Modelo:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"

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

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>com.autentia.tutoriales.icefaces</groupId>

<artifactId>Padre</artifactId>

<version>1.0-SNAPSHOT</version>

</parent>

<groupId>com.autentia.tutoriales.icefaces</groupId>

<artifactId>Modelo</artifactId>

<version>1.0-SNAPSHOT</version>
```

```
<packaging>ejb</packaging>

<name>Modelo</name>

<description>

Módulo encargado del modelo de negocio y de la persistencia de la aplicación.

Desarrollado con EJB 3.0.

</description>

<build>

<plugins>

<plugin>

<artifactId>maven-ejb-plugin</artifactId>

<configuration>

<ejbVersion>3.0</ejbVersion>

<archive>

<manifest>

<addClasspath>true</addClasspath>

</manifest>

</archive>

</configuration>

</plugin>

</plugins>

</build>

<dependencies>

<dependency>

<groupId>javax.persistence</groupId>

<artifactId>persistence-api</artifactId>

<version>1.0</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.transaction</groupId>

<artifactId>jta</artifactId>

<version>1.0.1B</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.activation</groupId>
```

```
<artifactId>activation</artifactId>
<version>1.1</version>
<scope>compile</scope>
</dependency>
<dependency>
<groupId>javax.ejb</groupId>
<artifactId>ejb-api</artifactId>
<version>3.0</version>
<scope>provided</scope>
</dependency>
<dependency>
<groupId>jboss</groupId>
<artifactId>jboss-ejb-api</artifactId>
<version>4.2.0.GA</version>
<scope>provided</scope>
</dependency>
</dependencies>
</project>
```

Por último, crearemos el pom.xml para la parte Web:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
<modelVersion>4.0.0</modelVersion>
<parent>
<groupId>com.autentia.tutoriales.icefaces</groupId>
<artifactId>Padre</artifactId>
<version>1.0-SNAPSHOT</version>
</parent>
<groupId>com.autentia.tutoriales.icefaces</groupId>
<artifactId>Web</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>war</packaging>
<name>Web</name>
<description>
Módulo encargado de la vista y el control. JSF RI e Icefaces
```

```
</description>
<dependencies>
<!-- Core J2EE -->
<dependency>
<groupId>javax.servlet</groupId>
<artifactId>servlet-api</artifactId>
<version>2.5</version>
<scope>provided</scope>
</dependency>
<dependency>
<groupId>javax.servlet.jsp</groupId>
<artifactId>jsp-api</artifactId>
<version>2.1</version>
<scope>provided</scope>
</dependency>
<dependency>
<groupId>javax.servlet</groupId>
<artifactId>jstl</artifactId>
<version>1.1.2</version>
</dependency>
<dependency>
<groupId>taglibs</groupId>
<artifactId>standard</artifactId>
<version>1.1.2</version>
</dependency>
<dependency>
<groupId>javax.el</groupId>
<artifactId>el-api</artifactId>
<version>1.0</version>
<scope>provided</scope>
</dependency>
<!-- JSF -->
<dependency>
<groupId>javax.faces</groupId>
<artifactId>jsf-api</artifactId>
<version>1.2_04-p01</version>
```

```
<scope>provided</scope>
</dependency>
<dependency>
<groupId>javax.faces</groupId>
<artifactId>jsf-impl</artifactId>
<version>1.2_04-p01</version>
<scope>provided</scope>
</dependency>
<!-- ICEFaces -->
<dependency>
<groupId>org.icefaces</groupId>
<artifactId>icefaces</artifactId>
<version>1.6.1</version>
</dependency>
<dependency>
<groupId>org.icefaces</groupId>
<artifactId>icefaces-comps</artifactId>
<version>1.6.1</version>
</dependency>
<dependency>
<groupId>org.icefaces</groupId>
<artifactId>icefaces-facelets</artifactId>
<version>1.6.1</version>
</dependency>
<dependency>
<groupId>backport-util-concurrent</groupId>
<artifactId>backport-util-concurrent</artifactId>
<version>3.1</version>
</dependency>
</dependencies>
</project>
```

2. **Usar el plugin m2Eclipse para gestionar los proyectos.**

   Configuraremos los tres proyectos para que sean gestionados por el plugin m2Eclipse: Si no lo tenéis, está explicado en: http://m2eclipse.codehaus.org/

Con esto acabamos la primera parte. En el próximo tutorial, configuraremos correctamente la parte del módulo Web para realizar un primer ejemplo.

- 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.

SOME RIGHTS RESERVED

This work is licensed under a Creative Commons Attribution-Noncommercial-No Derivative Works 2.5 License

.

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

soluciones reales para SU

Servicio de notificaciones:

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

Formulario de subcripción a novedades:

E-mail [ ] Aceptar

Otros Tutoriales Recomendados.

Nombre corto	Descripción
EJB 3.0: Resurrection	Este turorial nos va a presentar las nuevas funcionalidades que nos aportan los EJB 3.0.
Guía rápida de instalación de JBOSS Application Server 4.	En este manual veremos paso a paso la forma de instalar en tu equipo JBoss Application Server 4.
Planificar tareas en JBoss	En este tutorial os enseñaremos a planificar tareas periódicas con JBoss
MBeans y JBoss	Este tutorial tiene como finalidad familiarizarse con el servidor de aplicaciones JBoss y su API JMX
Manejo de JMS en JBOSS	Este tutorial nos dará un poco más de luz sobre JMS, un sistema de mensajería de SUN, sobre un servidor JBOSS con algunos ejemplos muy explicativos
EJB 3.0 y pruebas unitarias con Maven, JUnit y Embedded JBoss	En este tutorial Alejandro Pérez nos enseña como realizar test unitarios sobre EJB 3.0. Para ello se usará Maven, JUnit y Embedded JBoss
Despliegue gráfico de EJBs	Os mostramos como crear y desplegar de un modo gráfico un EJB de sesión el el servidor de aplicaciones de referencia de Sun
JBoss en cluster. Ejemplo de alta disponibilidad	Se pretende en este tutorial enseñar a todo aquel que esté interesado en este poderoso servidor de aplicaciones las distintas posibilidades que ofrece JBoss para manejar temas de clustering, alta disponibilidad etc... en aplicaciones J2EE.
Hibernate y las anotaciones de EJB 3.0	En este tutorial Alejandro Pérez nos muestra las ventajas que nos aporta Hibernate y las anotaciones de EJB 3.0

Interceptando un EJB en JBoss	En este tutorial os vamos a enseñar la arquitectura de EJBs en JBoss y a como modificarla, insertando un interceptor propio dentro de la cadena de interceptores del Proxy Cliente.

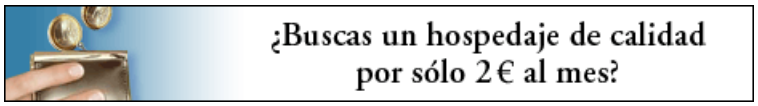
Ver todos los tutoriales

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.

Patrocinados por enredados.com .... Hosting en Castellano con soporte Java/J2EE