

# ¿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)

[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Tutoriales](#) | [Contacte](#)

¿Todavía gestionas tu empresa con hojas de cálculo? ¿No crees que habrá un sistema mejor? Ponemos a vuestra disposición TNTConcept, nuestra herramienta de gestión interna.

**CoNcept** Saber más .. <http://tntconcept.sourceforge.net>

### Tutorial desarrollado por:

[Raúl Expósito](#)

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](#).



[Catálogo de cursos](#)

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

Fecha de creación del tutorial: 2008-01-03

## Creación de un repositorio maven interno accesible por SSH

### 1. Introducción

Dado su potencial, cada vez es más frecuente encontrarse con proyectos que se apoyan en maven para ser desarrollados. En [Autentia](#) lo utilizamos y en [adictosaltrabajo](#) ya hemos hablado alguna vez sobre [él](#) en varias de sus facetas. Una de ellas es el repositorio de maven, entendiéndolo como el lugar donde maven encuentra las librerías y las dependencias que necesitamos para poder desarrollar el proyecto. En este tutorial vamos a centrarnos en la creación de un repositorio interno a nuestra empresa, un lugar gestionado por maven donde poder colocar librerías y dependencias de tal modo que sean accesibles por todos los

desarrolladores de nuestra empresa.

## 2. Problemática

Como ya hemos comentado una de las muchas cosas de las que maven se encarga es de descargar las librerías que necesitamos así como sus dependencias. Sin embargo, más tarde o más temprano podemos encontrarnos con alguno de estos efectos colaterales derivados del propio uso de esta tecnología:

- ¿qué ocurre si en nuestros proyectos necesitamos utilizar alguna librería desarrollada por un tercero que no existe en ningún repositorio público?
- ¿cómo hacemos para distribuir librerías o *wrappers* creados por nosotros mismos entre los desarrolladores de nuestra empresa?
- ¿qué ocurre si nuestra conexión a internet es lenta, intermitente, o preferimos darle otros usos más prioritarios?

## 3. Solución

En estos casos una buena idea es crear un repositorio de maven interno para los desarrolladores de nuestra empresa. De este modo solucionaremos los problemas presentados incluyendo todo aquello que vayamos a necesitar en nuestro propio repositorio.

## 4. Manos a la obra

Aunque la descarga de dependencias de un proyecto maven desde los repositorios se puede realizar a través de varias formas como http, https, ftp o scp voy a utilizar esta última por las siguientes razones:

1. Mantener, administrar y proteger un servidor ssh requiere menos esfuerzo que un servidor web o un servidor ftp, siempre a mi parecer.
2. Un servidor ssh maneja implícitamente la validación, y gracias a ella evitaremos que personas ajenas a nuestra empresa puedan acceder al repositorio.

Es por ello que para seguir este tutorial necesitaremos una máquina con un servidor ssh funcionando que pueda ser utilizada para montar nuestro repositorio.

### 4.1 Ficheros y directorios

Vamos a fijar 3 elementos que van a ser clave en la creación y posterior uso del repositorio:

- El directorio en la máquina que contiene el servidor ssh donde dejar el repositorio de maven. Por ejemplo `/var/maven/repository`.
- El fichero `pom.xml` del proyecto que estemos desarrollando, donde indicaremos entre otras cosas qué repositorios vamos a usar en el proyecto.
- El fichero `~/.m2/settings.xml`, donde configurar el usuario y la contraseña con los que vamos a acceder al repositorio de maven.

En este ejemplo el repositorio estará en la máquina servidorAutentiaMaven dentro del directorio `/var/maven/repository`

### 4.2 Creación del repositorio en el servidor

Basta con acceder a la máquina servidorAutentiaMaven y crear el directorio `/var/maven/repository/`

### 4.3 Edición del fichero `pom.xml`

Deberemos ir a la configuración de los repositorios que vamos a utilizar en el proyecto e incluir el nuestro de un modo similar a este:

```
<repositories>
  <repository>
    <id>servidorAutentiaMaven</id>
    <name>Repositorio de maven de Autentia</name>
    <url>scp://servidorAutentiaMaven/var/maven/repository</url>
  </repository>
</repositories>
```

Es muy importante que observéis el `<id>` y la `<url>`.

- La etiqueta `<id>` está relacionada con una configuración que haremos en `~/m2/settings.xml`
- En la etiqueta `<url>` indicamos el protocolo (scp), el nombre de la máquina (servidorAutentiaMaven) y el directorio donde encontrar el repositorio de maven (`/var/maven/repository`). Si por razones de seguridad o cualquier otra el puerto de escucha del servidor ssh no fuese el puerto por defecto, deberemos indicarlo de este modo:

```
<url>scp://servidorAutentiaMaven:<puerto>/var/maven/repository</url>
```

#### 4.4 Edición del fichero `~/m2/settings.xml`

Si no existiese tendríamos que crearlo y dejar un contenido similar a este, salvo que adaptado a nuestro caso particular:

```
<?xml version="1.0" encoding="UTF-8"?>
<settings>
  <localRepository>/home/raul/.m2/repository</localRepository>
  <servers>
    <server>
      <id>servidorAutentiaMaven</id>
      <username>raul</username>
      <password>miclavesupersecreta</password>
    </server>
  </servers>
</settings>
```

Si existiese, habría que modificarlo y ajustarlo. Hay que tener en cuenta lo siguiente:

- **El contenido de la etiqueta `<id>` debe coincidir con el de la etiqueta `<id>` del fichero `pom.xml` para saber a qué servidor estamos haciendo referencia** (podríamos tener más de uno)
- **En `<username>` y en `<password>` debemos poner el nombre de usuario y contraseña que vamos a utilizar al acceder a la maquina `servidorAutentiaMaven`**

#### 4.5 Probando que el repositorio funciona

Si hemos seguido los pasos anteriores podemos probar que el repositorio está bien creado y configurado con un sencillo 'mvn compile' desde la línea de comandos:

```
$ mvn compile
```

```
[INFO] Scanning for projects...
```

```
[INFO] -----
```

```
[INFO] Building project
```

```
[INFO] task-segment: [compile]
```

```
[INFO] -----
```

```
[INFO] artifact org.apache.maven.plugins:maven-compiler-plugin: checking for updates from servidorAutentiaMaven
```

```
The authenticity of host 'servidorAutentiaMaven' can't be established.
```

```
RSA key fingerprint is ef:ae:b0:0d:ea:c5:40:33:dc:2f:f9:e3:a5:4c:c4:24.
```

```
Are you sure you want to continue connecting? (yes/no): yes
```

```
[INFO] [resources:resources]
```

```
[INFO] Using default encoding to copy filtered resources.
```

```
Downloading: scp://servidorAutentiaMaven/var/maven/repository/javax/transaction/jta/1.0.1B/jta-1.0.1B.jar
```

```
Downloading: http://repo1.maven.org/maven2/javax/transaction/jta/1.0.1B/jta-1.0.1B.jar
```

```
Downloading: scp://servidorAutentiaMaven/var/maven/repository/com/sun/jdmk/jmxtools/1.2.1/jmxtools-1.2.1.jar
```

```
Downloading: https://maven-repository.dev.java.net/nonav/repository/com.sun.jdmk/jars/jmxtools-1.2.1.jar
```

```
Downloading: http://repo1.maven.org/maven2/com/sun/jdmk/jmxtools/1.2.1/jmxtools-1.2.1.jar
```

```
Downloading: scp://servidorAutentiaMaven/var/maven/repository/com/sun/jmx/jmxri/1.2.1/jmxri-1.2.1.jar
```

```
Downloading: https://maven-repository.dev.java.net/nonav/repository/com.sun.jmx/jars/jmxri-1.2.1.jar
```

```
Downloading: http://repo1.maven.org/maven2/com/sun/jmx/jmxri/1.2.1/jmxri-1.2.1.jar
```

Downloading: **scp://servidorAutentiaMaven/var/maven/repository/javax/jms/jms/1.1/jms-1.1.jar**

Downloading: <https://maven-repository.dev.java.net/nonav/repository/javax/jms/jars/jms-1.1.jar>

Downloading: <http://repo1.maven.org/maven2/javax/jms/jms/1.1/jms-1.1.jar>

Como veis conecta con servidorAutentiaMaven y trata de bajar las dependencias del proyecto.

## 5. Subiendo librerías de terceros al servidor

Hay veces que queremos usar librerías de terceros que no se encuentran en ningún repositorio remoto y que deberemos importar a nuestro repositorio local. Por ejemplo, las librerías de jta.

### [5.1] JTA

Si intentamos utilizarlas y no las tenemos en nuestro repositorio local nos aparecerá un error como el siguiente:

```
[INFO] -----
[ERROR] BUILD ERROR
[INFO] -----
[INFO] Failed to resolve artifact.
```

Missing:

-----

1) javax.transaction:jta:jar:1.0.1B

Try downloading the file manually from:

<http://java.sun.com/products/jta>

Then, install it using the command:

```
mvn install:install-file -DgroupId=javax.transaction -DartifactId=jta \
-Dversion=1.0.1B -Dpackaging=jar -Dfile=/path/to/file
```

Alternatively, if you host your own repository you can deploy the file there: `mvn deploy:deploy-file -DgroupId=javax.transaction -DartifactId=jta \`

```
-Dversion=1.0.1B -Dpackaging=jar -Dfile=/path/to/file \
```

```
-Durl=[url] -DrepositoryId=[id]
```

Para resolver este problema vamos a instalar las librerías en el repositorio de maven que hemos creado para que cualquier desarrollador de la empresa que las necesite pueda utilizarlas fácilmente. Maven ya nos da instrucciones para ello:

Try downloading the file manually from:

<http://java.sun.com/products/jta>

[...]

if you host your own repository you can deploy the file there:

```
mvn deploy:deploy-file -DgroupId=javax.transaction -DartifactId=jta \
```

```
-Dversion=1.0.1B -Dpackaging=jar -Dfile=/path/to/file \
```

```
-Durl=[url] -DrepositoryId=[id]
```

Descargamos la librería de <http://java.sun.com/products/jta> y ejecutamos un comando parecido a este:

```
mvn deploy:deploy-file -DgroupId=javax.transaction -DartifactId=jta -Dversion=1.0.1B -Dpackaging=jar
-Dfile=/home/raul/jta-1_0_1B-classes.zip -Durl=scp://servidorAutentiaMaven/var/maven/repository
-DrepositoryId=servidorAutentiaMaven
```

Eso si, deberemos tener permisos de escritura en el directorio `/var/maven/repository` de la máquina servidorAutentiaMaven para poder instalar la librería

### [5.2] JMXRI y JMXTools

Nos vamos a <http://java.sun.com/products/JavaManagement/download.html> y descargamos el enlace 'JMX 1.2.1 Reference Implementation'.

Dentro del zip que hemos descargado, en el directorio `lib/`, encontraremos dos jar: `jmxri.jar` y `jmxtools.jar`. Los descomprimos y

los instalamos en el repositorio:

```
mvn deploy:deploy-file -DgroupId=com.sun.jdmk -DartifactId=jmxtools -Dversion=1.2.1 -Dpackaging=jar
-Dfile=/home/raul/jmxtools.jar -Durl=scp://servidorAutentiaMaven/var/maven/repository -DrepositoryId=servidorAutentiaMaven

mvn deploy:deploy-file -DgroupId=com.sun.jmx -DartifactId=jmxri -Dversion=1.2.1 -Dpackaging=jar
-Dfile=/home/raul/jmx-1_2_1-ri.zip -Durl=scp://servidorAutentiaMaven/var/maven/repository
-DrepositoryId=servidorAutentiaMaven
```

### [5.3] JMS

De un modo parecido al anterior nos vamos a <http://java.sun.com/products/jms/docs.html> y descargamos 'Download the version 1.1 API Documentation, Jar and Source'

Dentro del zip que hemos descargado, en el directorio lib/, encontraremos dos jar: javax.jms.jar y jms.jar. Descomprimos el javax.jms.jar y lo instalamos:

```
mvn deploy:deploy-file -DgroupId=javax.jms -DartifactId=jms -Dversion=1.1 -Dpackaging=jar -Dfile=/home/raul/javax.jms.jar
-Durl=scp://servidorAutentiaMaven/var/maven/repository -DrepositoryId=servidorAutentiaMaven
```

## 6. Conclusiones

Si utilizais maven para desarrollar proyectos la utilidad de un repositorio es clara, ya que os permitirá tener localizadas librerías de terceros que no estén en repositorios públicos y os permitirá poder compartir librerías o componentes creadas por vosotros mismos entre proyectos de una manera sencilla. Además como habeis podido ver a lo largo del tutorial crear y gestionar el repositorio es bastante sencillo.

Si incluir maven en vuestros proyectos es una buena idea, ¿no es tener un repositorio propio un buen complemento a esta buena idea?.

Espero que os sea de utilidad.

- 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](#).



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](mailto:info@autentia.com)

(Block)



## Servicio de notificaciones:

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

Formulario de subcripción a novedades:

E-mail

## Otros Tutoriales Recomendados.

Nombre corto	Descripción
<a href="#">Acceso seguro a CVS a través de SSH</a>	Os mostramos como securizar los accesos a CVS a través de SSH, utilizando herramientas gratuitas
<a href="#">Desarrollo de Plugins para Maven</a>	En este tutorial vamos a hacer un ejemplo práctico de construcción de un plugin para Maven (es la herramienta de gestión de la construcción).
<a href="#">Q, un plugin de Eclipse para gestionar Maven 2</a>	en este tutorial aprendemos a instalar y usar Q, un plugin de Eclipse para gestionar Maven 2
<a href="#">Pruebas de integración con Maven</a>	Este tutorial nos muestra un ejemplo para lanzar las pruebas de integración "engañando" a Maven para que no se lanzen en la fase de test teniendo únicamente un módulo para ambas
<a href="#">EJB 3.0, un ejemplo práctico con Maven y JBoss</a>	Este tutorial presenta un ejemplo sencillo donde se verá como desarrollar EJBs de sesión y de entidad, inyección de dependencias, llamar a los EJBs desde una aplicación Web, definición de un DataSource, y como configurarlo y hacerlo funcionar en JBoss, y
<a href="#">Crear el sitio web de documentación del proyecto con Maven Site</a>	En este tutorial vamos a ver cómo crear los sitios web de documentación del proyecto con Apache Maven.
<a href="#">Como generar con Maven un histórico de cambios del proyecto</a>	Como generar con Maven un histórico de cambios del proyecto.
<a href="#">EJB 3.0 y pruebas unitarias con Maven, JUnit y Embedded JBoss</a>	En este tutorial Alejandro Pérez nos enseña como realizar test unitarios sobre EJB 3.0. Para ello se usará Maven, JUnit y Embedded JBoss
<a href="#">Proyecto con JSF Myfaces, Maven y Eclipse</a>	En este tutorial vamos a aprender a construir una aplicación básica JSF (Java Server Pages) utilizando el Maven 2.0 y las bibliotecas de MyFaces. Lo mejor de todo es que para crear el ejemplo no vamos a programar ni una línea.
<a href="#">Crear un repositorio remoto y como hacer una 'release' con varios proyectos en Maven y Eclipse</a>	En este tutorial vamos a explicar como podemos trabajar teniendo varios proyectos relacionados en Maven y en Eclipse
<a href="#">Ver todos los tutoriales</a>	

### 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](mailto:rcanales@adictosaltrabajo.com) para su resolución.

[Patrocinados por enredados.com ..... Hosting en Castellano con soporte Java/J2EE](#)

