Normativa de Desarrollo de Maven y Archiva

Fecha: 13/05/2013

Referencia:

EJIE S.A.

Mediterráneo, 3

01010 Vitoria-Gasteiz

Posta-kutxatila / Apartado: 809

01080 Vitoria-Gasteiz

Tel. 945 01 73 00*

Fax. 945 01 73 01

www.EJIE.es

Este documento es propiedad de EJIE, S.A. y su contenido es confidencial. Este documento no puede ser reproducido, en su totalidad o parcialmente, ni mostrado a otros, ni utilizado para otros propósitos que los que han originado su entrega, sin el previo permiso escrito de EJIE, S.A.. En el caso de ser entregado en virtud de un contrato, su utilización estará limitada a lo expresamente autorizado en dicho contrato. EJIE, S.A. no podrá ser considerada responsable de eventuales errores u omisiones en la edición del documento.

Control de documentación

Título de documento: **Normativa de Desarrollo Maven en WLS11**

Histórico de versiones

Código:	Versión:	Fecha:	Resumen de cambios:
	01	09.03.2011	Primera versión
	02	13.05.2013	Parámetro nuevo para traspasos en anthill pro en tareas de obtención de dependencias Nuevo repositorio de artefactos de pruebas.
	03	30.10.2013	Nuevo build en local, con tarea cleanLib.

Cambios producidos desde la última versión

13/05/2013

Control de difusión

Responsable: Ander Martínez

Aprobado por: Ander Martínez

Firma:	Fecha:

Distribución: Interna

Referencias de archivo

Autor: Andrés Sáenz-Otalora Garmendia

Nombre archivo: Normativa Desarrollo Maven y Archiva.doc

Localización:

Contenido

1. Introducción

El presente documento presenta las herramientas disponibles para el desarrollador que quiera hacer uso de los repositorios de Maven en Archiva de EJIE para aplicaciones en Weblogic 11, normado su uso.

Maven es una herramienta de software para la gestión y construcción de proyectos.
Se basa en la utilización de ficheros POM (Project Object Model) que describen el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Realiza tareas de compilación del código y empaquetado, gestiona dependencias, etc.

En EJIE las tareas de empaquetado y compilación se gestionan vía tareas ant. Maven se utilizará solamente para la gestión de dependencias y publicación de artefactos en los repositorios Maven.

EJIE pone a disposición del desarrollador repositorios Maven 2 en el entorno de desarrollo y tareas ant que permitirán gestionar dependencias así como publicar artefactos. Además expone dichos repositorios vía http para obtención de artefactos (usando tareas ant en el cliente) y vía una aplicación web (Archiva). Esta aplicación permite realizar búsquedas de artefactos, y si se es administrador de tu propio repositorio subirlos, borrarlos, etc.

2. Herramientas en Servidor de Desarrollo

En el entorno de desarrollo se ha implantado un gestor de repositorios Maven llamado Archiva que expondrá los diferentes repositorios al desarrollador. En el desarrollo de una aplicación se podrá hacer uso de las tareas ant de maven en el servidor de desarrollo para bajarse dependencias existentes en los repositorios Maven de EJIE, así como publicar artefactos en ellos.

2.1. Repositorios Maven 2 en EJIE

Dentro del gestor de repositorios de Archiva se han creado los siguientes repositorios de uso general:

- **soportadas**: Repositorio que contendrá aquellas librerías a cuyo uso directo se dará soporte por parte del dpto. de Consultoría de Áreas de Conocimiento, en adelante CAC.
  URL: http://www.otc.ejiedes.net/archiva/repository/soportadas/
- **sopTesting**: Repositorio que contendrá aquellas librerías de testing a cuyo uso directo se dará soporte por parte del dpto. de Consultoría de Áreas de Conocimiento, en adelante CAC.
  URL: http://www.otc.ejiedes.net/archiva/repository/sopTesting/
- **soportadasDep**: Repositorio que contendrá aquellas librerías dependientes de las soportadas y sopTesting cuyo uso directo no estará soportado por parte de CAC.
  URL: http://www.otc.ejiedes.net/archiva/repository/soportadasDep/
- **repoAplicPublicas**: Repositorio que contendrá aquellas librerías que las aplicaciones publiquen para que sean usados por otras.
  URL: http://www.otc.ejiedes.net/archiva/repository/repoAplicPublicas/
- **repoEJIE**: Repositorio que engloba a los anteriores para obtener dependencias
  URL: http://www.otc.ejiedes.net/archiva/repository/repoEJIE/

A parte de estos repositorios generales existirán **repositorios privados**. Aquellas factorías de software que requieran publicar artefactos para un uso interno departamental o de un ámbito que abarque a un grupo de aplicaciones grande podrán solicitar su propio repositorio.
Un ejemplo podría ser la creación de un repositorio Maven para el departamento de educación. Para ello se deberá realizar una solicitud a CAC de nuevo repositorio especificando el nombre de la factoría. Dicho nombre deberá ser un nombre formato CamelCase. La solicitud de nuevo repositorio se tendrá que evaluar en fase -1 junto con el dpto. de CAC.

Ejemplo:

Nombre Repositorio: *repoEducacion.*

CAC devolverá dos URLs: una del repositorio nuevo creado y otra de la URL del repositorio de grupo (engloba los repositorios genéricos citados anteriormente más el nuevo creado) que sirve para bajarse las dependencias y el usuario y password. Dicho usuario y password es necesario para bajarse las dependencias, así como para logarse en Archiva para gestionar el repositorio privado. Se proporcionan tambien las URLs internas de ambos repositorios que se necesitarán para configurar el entorno de desarrollo de las tareas ant.

Ejemplo:

URL externa Repositorio Grupo: http://www.otc.ejiedes.net/archiva/repository/repoEducacionRep/
URL interna Repositorio Grupo:
http://otc.tomcat.ejiedes.net:8090/archiva/repository/repoEducacionRep/
URL externa Repositorio privado : http://www.otc.ejiedes.net/archiva/repository/repoEducacion/
URL interna Repositorio privado :
http://otc.tomcat.ejiedes.net:8090/archiva/repository/repoEducacion/

Usuario: *repoEducacion*
Password: *123asd34*

2.2. Tareas Ant en servidor de Desarrollo

En el apartado anterior se citan los diferentes repositorios que estarán albergados en EJIE. Sobre ellos se podrá realizar diferentes actuaciones. Las tareas ant de servidor nos permitirán publicar artefactos en ellos y bajar dependencias de ellos.

Se han creado en el servidor 4 nuevas tareas ant:

- getDependenciesAppArchiva. Baja las dependencias especificadas en los ficheros pom para un aplicación EAR.

- getDependenciesLibArchiva. Baja las dependencias especificadas en los ficheros pom para una aplicación JAR.

- publishPublicLib. Publica un artefacto generado en el repositorio de publicaciones compartidas (verifica que el groupid sea del tipo com.ejie.*).
  Si no se especifican parámetros publica el artefacto como en la tarea de publicación existente actualmente.

- publishPrivateLib. Publica un artefacto generado en el repositorio privado de la factoría (verifica que el groupid sea del tipo com.ejie.*).
  Si no se especifican parámetros publica el artefacto como en la tarea de publicación existente actualmente.

2.2.1. getDependenciesAppArchiva

Esta tarea baja las dependencias especificadas en los ficheros pom ubicados en el directorio /aplic/bbb/tmp/deps: *pom_app.xml* y los diversos *pom_*<nombreWar>*.xml*.
El fichero pom_app.xml contendrá las dependencias de la apliacación EAR y las descargará en la ruta */aplic/bbb/src/bbbEAR/APP-INF/lib*.
Si existen módulos War, por cada uno de ellos existirá un *pom_*<nombreWar>*.xml*. Sus dependencias las bajará a la ruta */aplic/bbb/src/bbbEAR/wars/<nombreWar>/WEB-INF/lib* de cada módulo web.

Permite un parámetro opcional *DconfEnv* que podrá tener los valores "/" si el desarrollo está en el entorno de desarrollo productivo y "/softbase_ejie/" si se desarrolla en el entorno de softbase. Este parámetro se ha de pasar obligatoriamente en los traspasos con Anthill Pro.

Ej: ant getDependenciesAppArchiva –DconfEnv=/

2.2.2. getDependenciesLibArchiva

Esta tarea baja las dependencias especificadas en el fichero pom /aplic/bbb/tmp/deps/*pom_lib.xml*. Este fichero contiene las dependencias de la librería. La tarea ant descargará las dependencias en la ruta */aplic/bbb/src/bbbShLibClasses/lib*.

Permite un parámetro opcional *DconfEnv* que podrá tener los valores “/” si el desarrollo está en el entorno de desarrollo productivo y “/softbase_ejie” si se desarrolla en el entorno de softbase. Este parámetro se ha de pasar obligatoriamente en los traspasos con Anthill Pro.

Ej: ant getDependenciesLibArchiva –DconfEnv=/

2.2.3. publishPublicLib

Publica una librería en el repositorio de librerías compartidas de EJIE *repoAplicPublicas*. Esta tarea si no se le pasa parámetros publicará la librería generada por la tarea *createShLibJar*.y necesitará tener configurado correctamente el fichero pom_lib.xml en /aplic/bbb/tmp/deps.

Ej: ant publishPublicLib

En el caso de que se pretenda subir otro artefacto generado, como pudieran ser clientes generados de EJBs, etc se suministrarán parámetros adicionales:

- pomfile: fichero pom del artefacto
- jarfile: artefacto jar a publicar.

Ej: ant publishPrivateLib -Dpomfile=/aplic/tmp/deps/pom_stubs.xml -Djarfile=../bbbStubsEJBs.jar

2.2.4. publishPrivateLib

Publica una librería en el repositorio de privado de la factoría de software. Esta tarea si no se le pasa parámetros publicará la librería generada por la tarea *createShLibJar*.y necesitará tener configurado correctamente el fichero pom_lib.xml en /aplic/bbb/tmp/deps.

Ej: ant publishPrivateLib

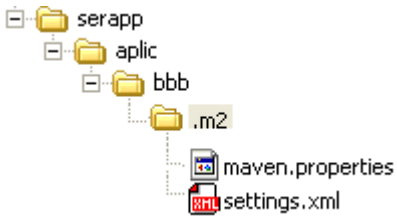
En el caso de que se pretenda subir otro artefacto generado, como pudieran ser clientes generados de EJBs, etc se suministrarán parámetros adicionales:

- pomfile: fichero pom del artefacto
- jarfile: artefacto jar a publicar.

Ej: ant publishPrivateLib -Dpomfile=/aplic/bbb/tmp/deps/pom_stubs.xml -Djarfile=../bbbStubsEJBs.jar

2.2.5. Estructura Directorios necesaria

Una aplicación que necesite usar las tareas ant de Maven necesitará crear la siguiente estructura de directorios:



1. Crear la carpeta /entorno/aplic/bbb/.m2
2. Creación del fichero /entorno/aplic/bbb/.m2/maven.properties con el contenido siguiente si se ha solicitado el uso de un repositorio privado (hay que utilizar las URLs internas proporcionadas):

```
repoURL=http://otc.tomcat.ejiedes.net:8090/archiva/repository/repoEduc
          acionRep/
```

```
publishRepoPrivate=http://otc.tomcat.ejiedes.net:8090/archiva/reposito
          ry/repoEducacion/
```

En el caso de no haber solicitado un repositorio privado:

```
repoURL=${repoEJIE}
```

3. Creación del fichero /entorno/aplic/bbb/.m2/settings.xml con el contenido siguiente:

Si se ha solicitado el uso de un repositorio privado se deberá introducir los datos del usuario creado por CAC. Así en el elemento username se meterá el usuario nuevo introducido y en el password su clave.

```
<settings>
  <proxies>
  </proxies>
  <servers>
    <server>
        <id>repo</id>
        <username>repoEducacion</username>
        <password>123asd34</password>
      </server>
  </servers>
<mirrors>
  <mirror>
      <id>m1</id>
      <name>Mirror1</name>
       <url>http://otc.tomcat.ejiedes.net:8090/archiva/repository/
       repoEducacionRep/</url>
      <mirrorOf>central</mirrorOf>
  </mirror>
  <mirror>
      <id>m2</id>
      <name>Mirror2</name>
       <url>http://otc.tomcat.ejiedes.net:8090/archiva/repository/
       repoEducacionRep/</url>
      <mirrorOf>repoGet</mirrorOf>
  </mirror>
  <mirror>
      <id>m3</id>
      <name>Mirror3</name>
       <url>http://otc.tomcat.ejiedes.net:8090/archiva/repository/
       repoEducacionRep/</url>
      <mirrorOf>ejie</mirrorOf>
  </mirror>
  </mirrors>
  <profiles>
  </profiles>
</settings>
```

URLs interna del repositorio de Grupo

Si no se usa un repositorio privado el archivo deberá tener lo siguiente:

```
<settings>
  <proxies>
```

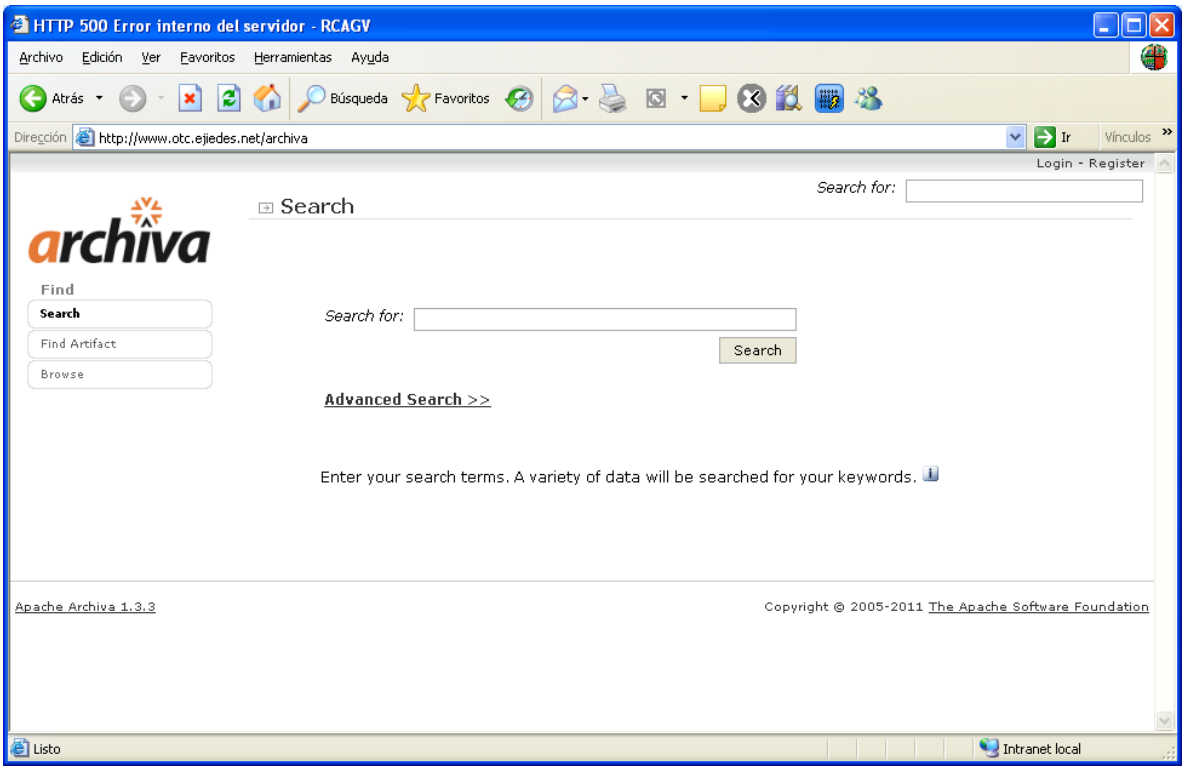
```
  </proxies>
  <servers>
    <server>
        <id>repo</id>
        <username></username>
        <password></password>
      </server>
  </servers>
<mirrors>
  </mirrors>
  <profiles>
  </profiles>
</settings>
```

2.3. Aplicación web Archiva

Los desarrolladores podrán acceder a la aplicación web Archiva para realizar consultas sobre las librerías soportadas en EJIE, o en caso de que dispongan de un repositorio propio subir o borrar librerías.

La URL es http://www.otc.ejiedes.net/archiva.

Si se accede a esta página se muestra la ventana siguiente:



No hace falta logarse en dicha aplicación ya que esta habilitado el usuario invitado. De esta forma permitirá realizar consultas de librerías a utilizar.

La aplicación nos permitirá realizar las siguientes funcionalidades:

- Búsquedas y descarga de artefactos
- Navegación en el repositorio
- Subir artefactos (repositorio privado)
- Borrado artefactos (repositorio privado)
- Auditoria del repositorio

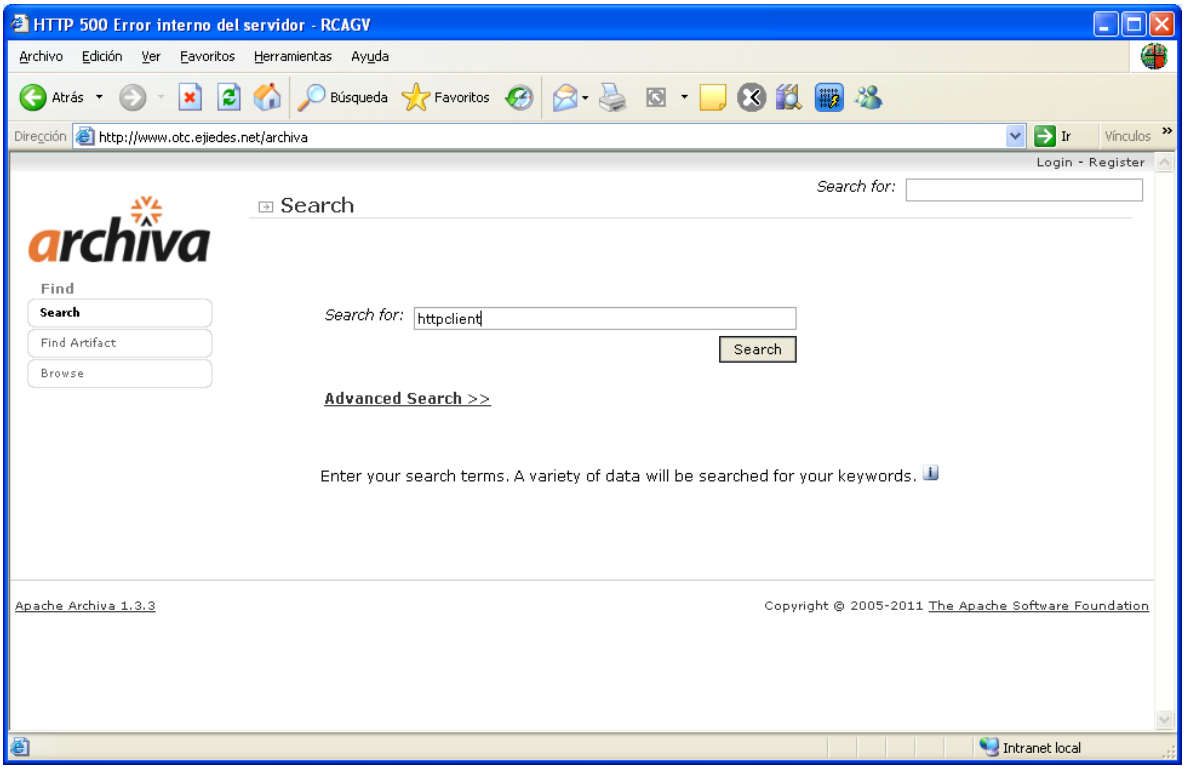
2.4. Búsquedas de artefactos

Archiva nos permite realizar búsquedas de artefactos de 3 formas distintas: búsqueda sencilla, avanzada y una búsqueda a partir de un artefacto en local.

2.4.1. Búsqueda sencilla

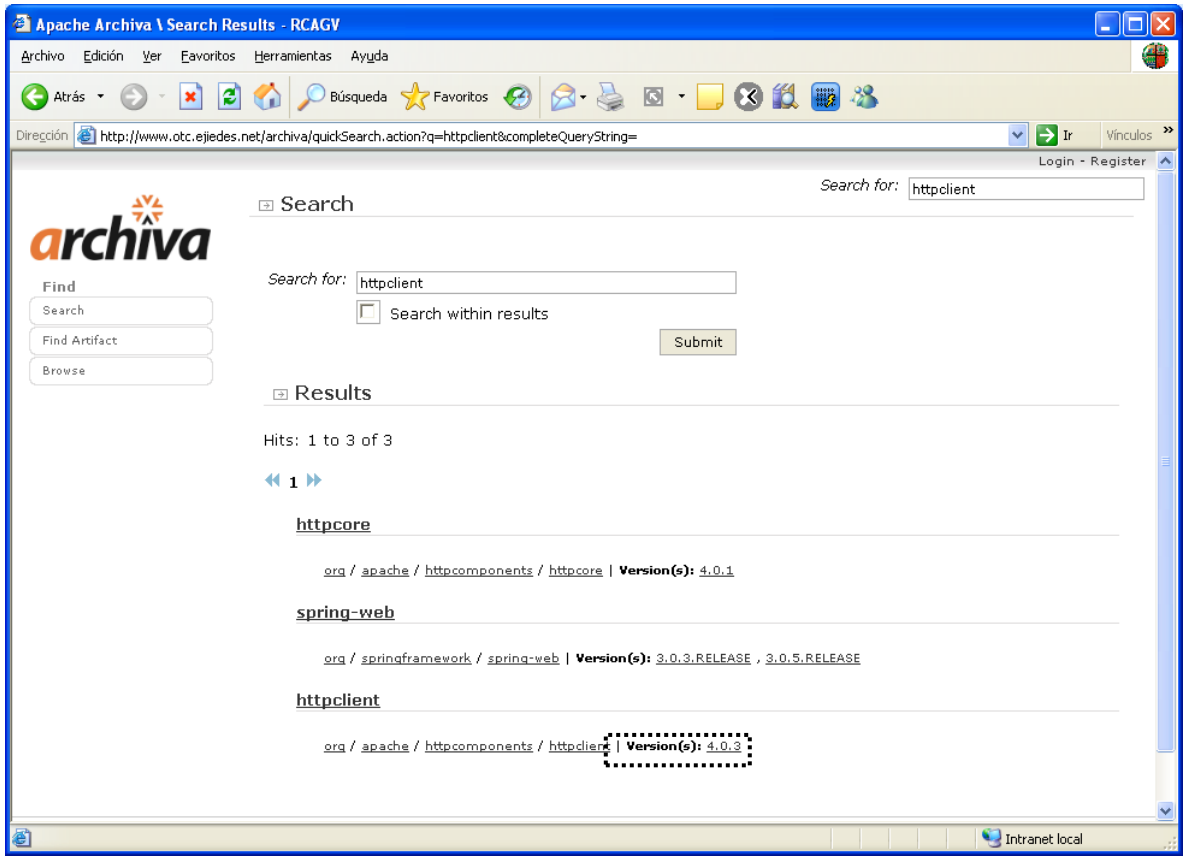
La búsqueda sencilla nos buscará el literal en los repositorios a los que se tenga permiso de observador. Cuando no se autentica un usuario tendrá permiso de observador sobre todos los repositorios. Por ello se recomienda realizar utilizar la búsqueda avanzada sobre el repositorio de soportadas si va a buscar una librería de terceros o sobre el repositorio de repoAplicPublicas en caso de buscar una librería de una aplicación de EJIE.

Se introduce el texto a buscar tal y como aparece en la pantalla siguiente:



El texto se buscará en los ficheros pom de cada artefacto pudiendo aparecer en dependencias así como en el nombre del artefacto o en elementos del fichero.

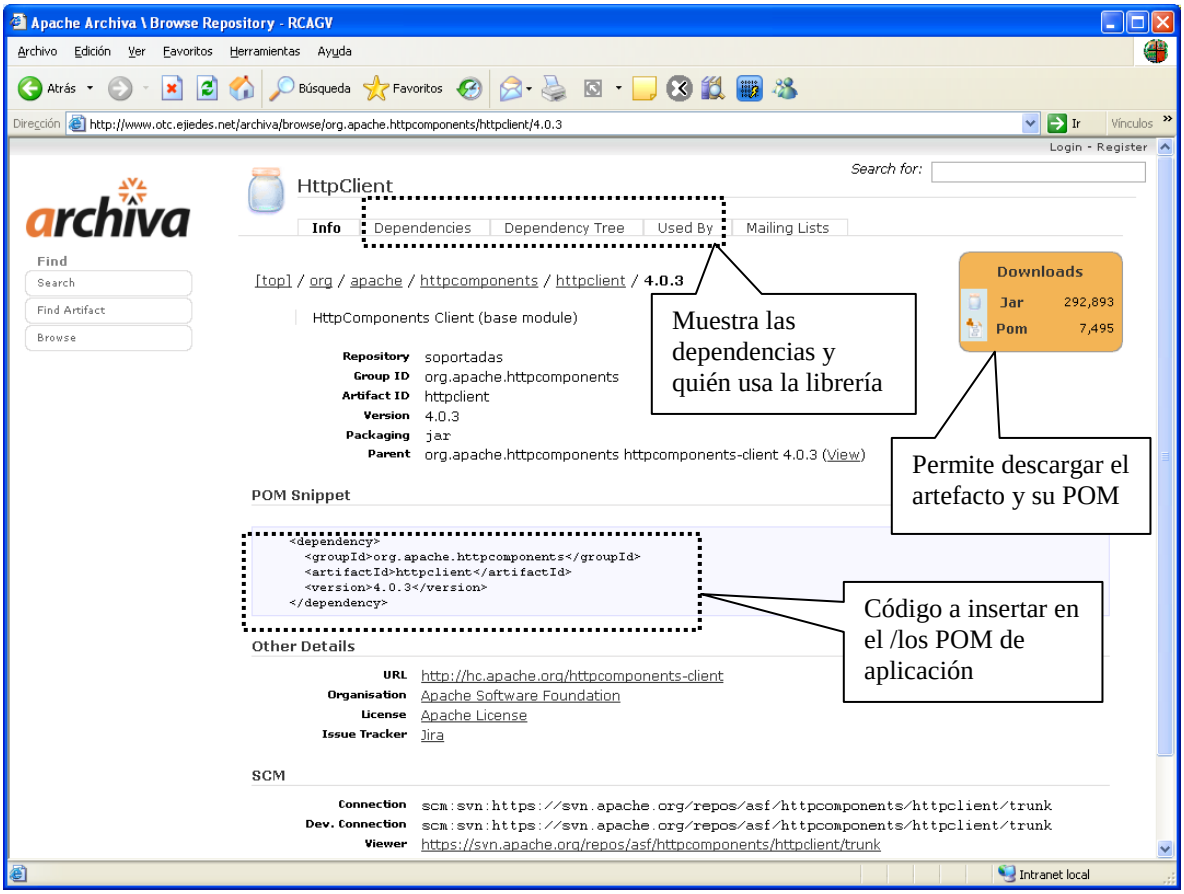
Se muestran los resultados de la búsqueda



Se selecciona el artefacto que pretendemos descargar o visualizar seleccionando sobre la versión concreta.

Nos mostrará la información disponible en el pom de la librería en un formato humano y utilidades extra:

-Visualización de datos del pom (Group ID, Artifact ID, Version, Packaging, etc). Nos muestra las dependencias de la librería en formato humano, así como que librerías la usan.
-Download artefacto, su pom correspondiente
-POM Snippet: para pegar en el/los pom de nuestra aplicación.

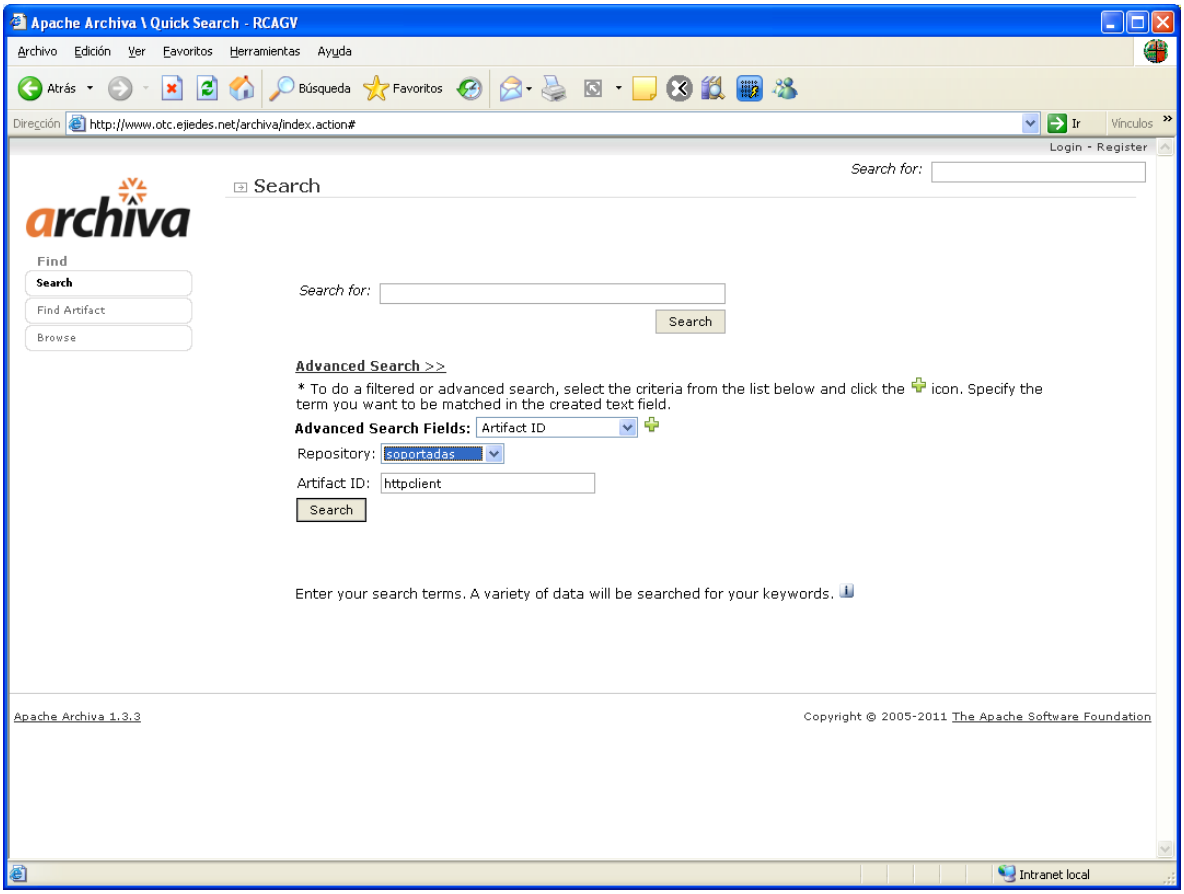


2.4.2. Búsqueda avanzada

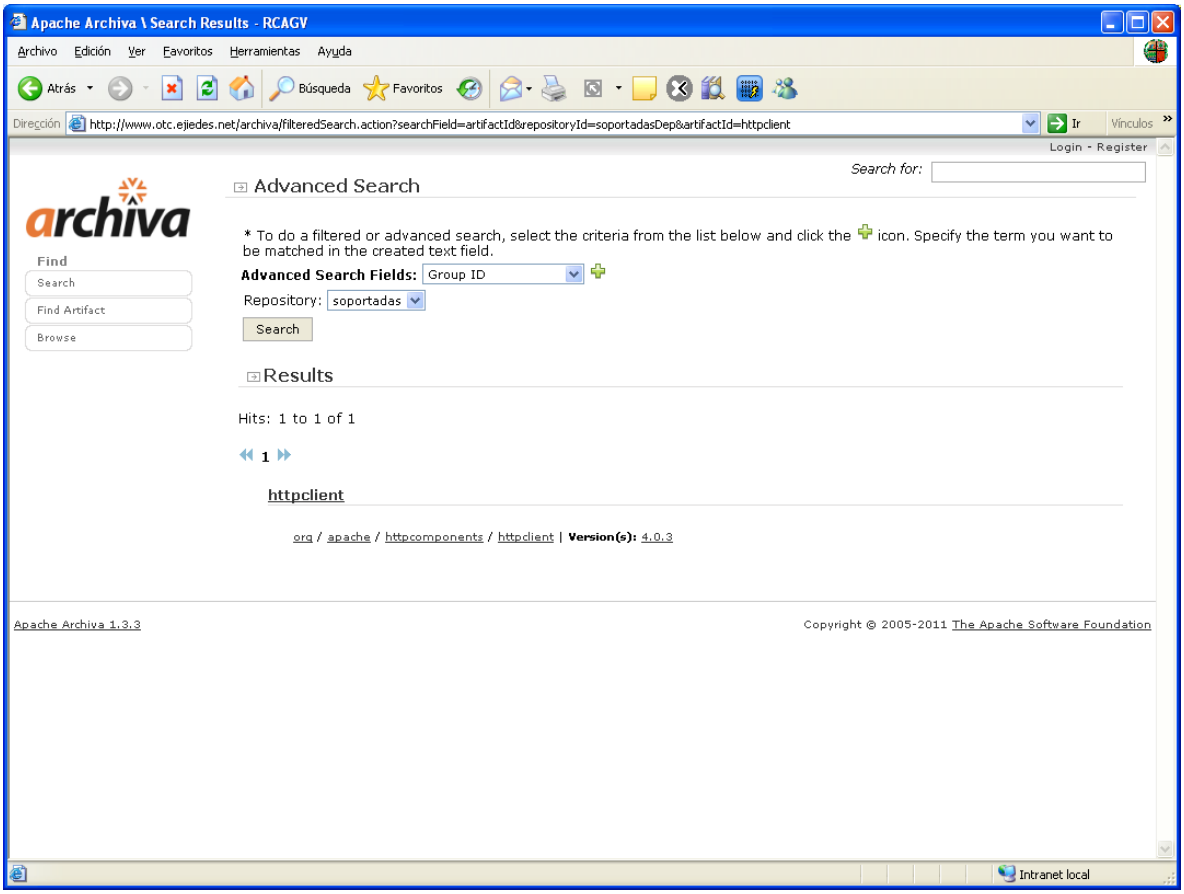
La búsqueda avanzada nos permite filtrar por repositorios y por campos concretos del pom.

Un ejemplo prático sería buscar por ejemplo cual es la librería soportada en EJIE de httpclient.
Para ello deberemos filtrar por lo siguientes campos:

- Artefact ID = httpclient
- Repository = soportadas. Deberemos seleccionar siempre este repositorio para comprobar las librerías de terceros que da soporte EJIE.



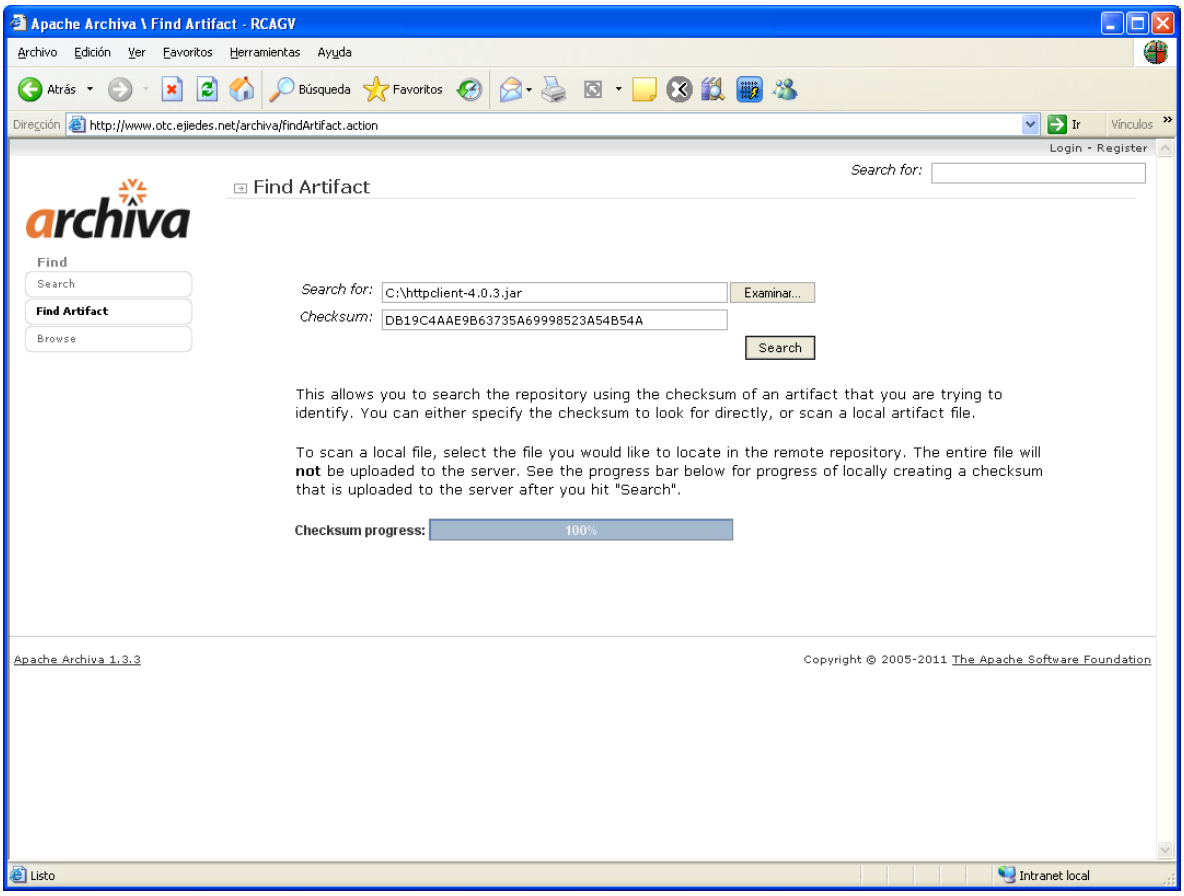
A continuación nos saldrán los resultados de la búsqueda.



A partir de aquí para visualizar el artefacto que deseemos funciona de manera idéntica a la búsqueda sencilla.
Mediante la búsqueda avanzada se tendrán que buscar las librerías soportadas por EJIE.

2.4.3. Búsqueda a partir de un artefacto local

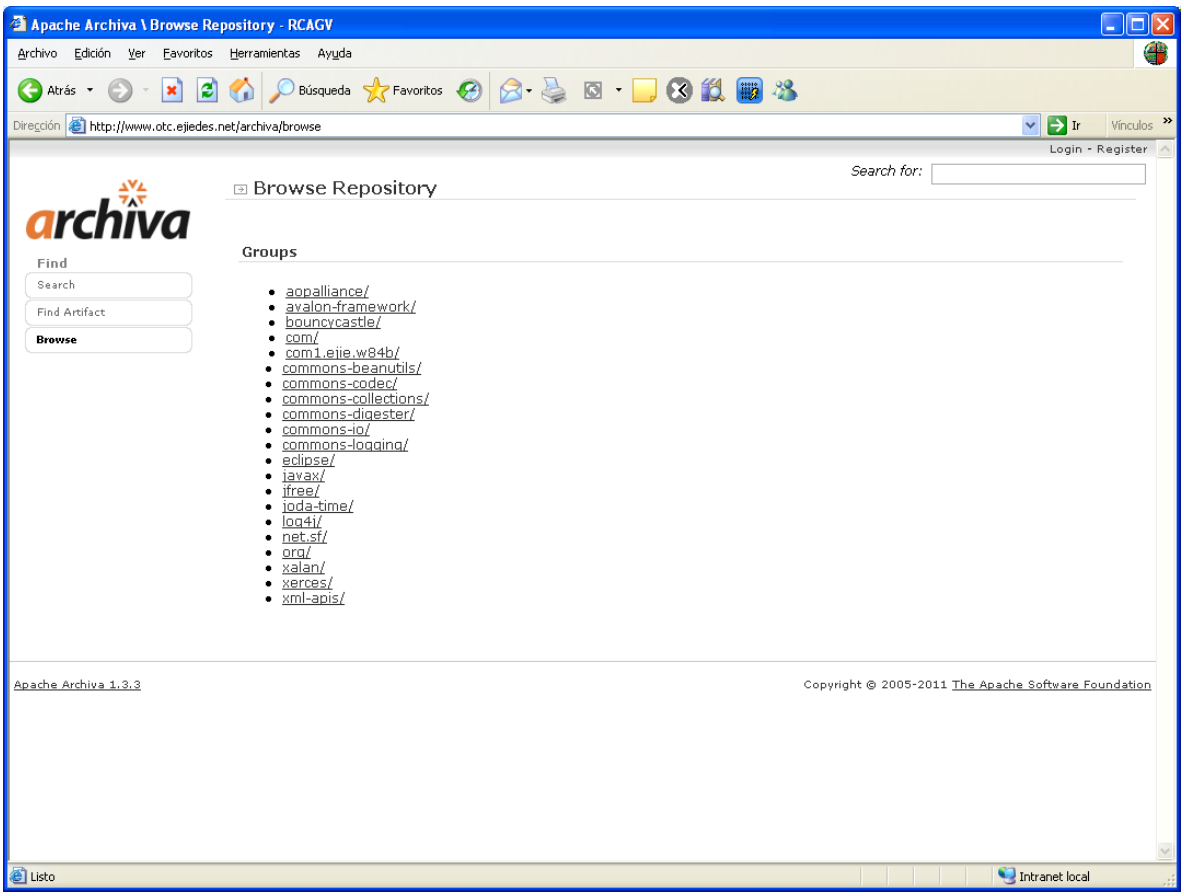
Esta búsqueda permite al desarrollador comprobar si teniendo un artefacto en local ver si existe este ya en los repositorios a los que tiene acceso. De esta forma seleccionando un archivo en local te calcula un hash de éste y realiza la búsqueda a partir de el.



2.5. Navegación en el repositorio

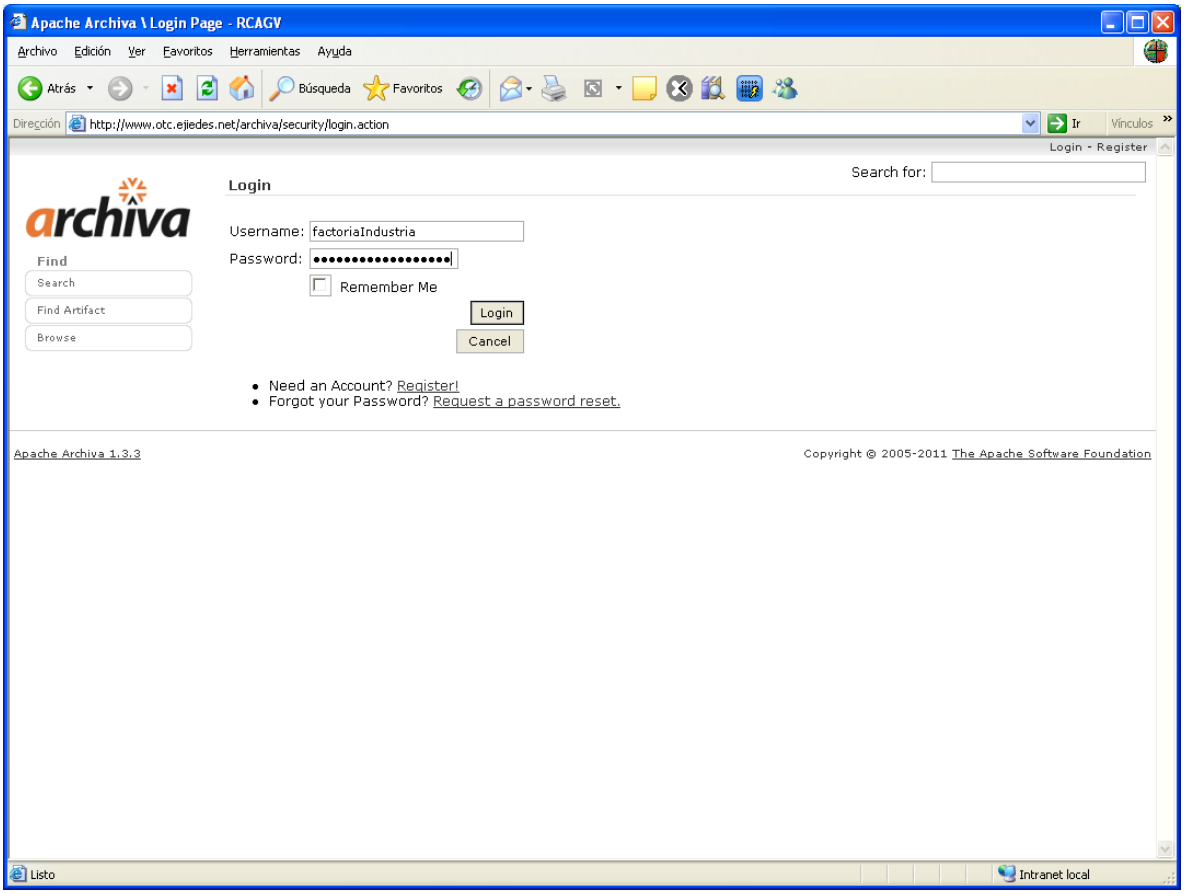
Esta opción permite navegar en la estructura de directorios de los repositorios a los que se tiene acceso. Se fusionarán todos los repositorios a los que se tenga acceso en una estructura única.

Se mostrará la siguiente pantalla.



2.6. Subir artefactos (repositorio privado)

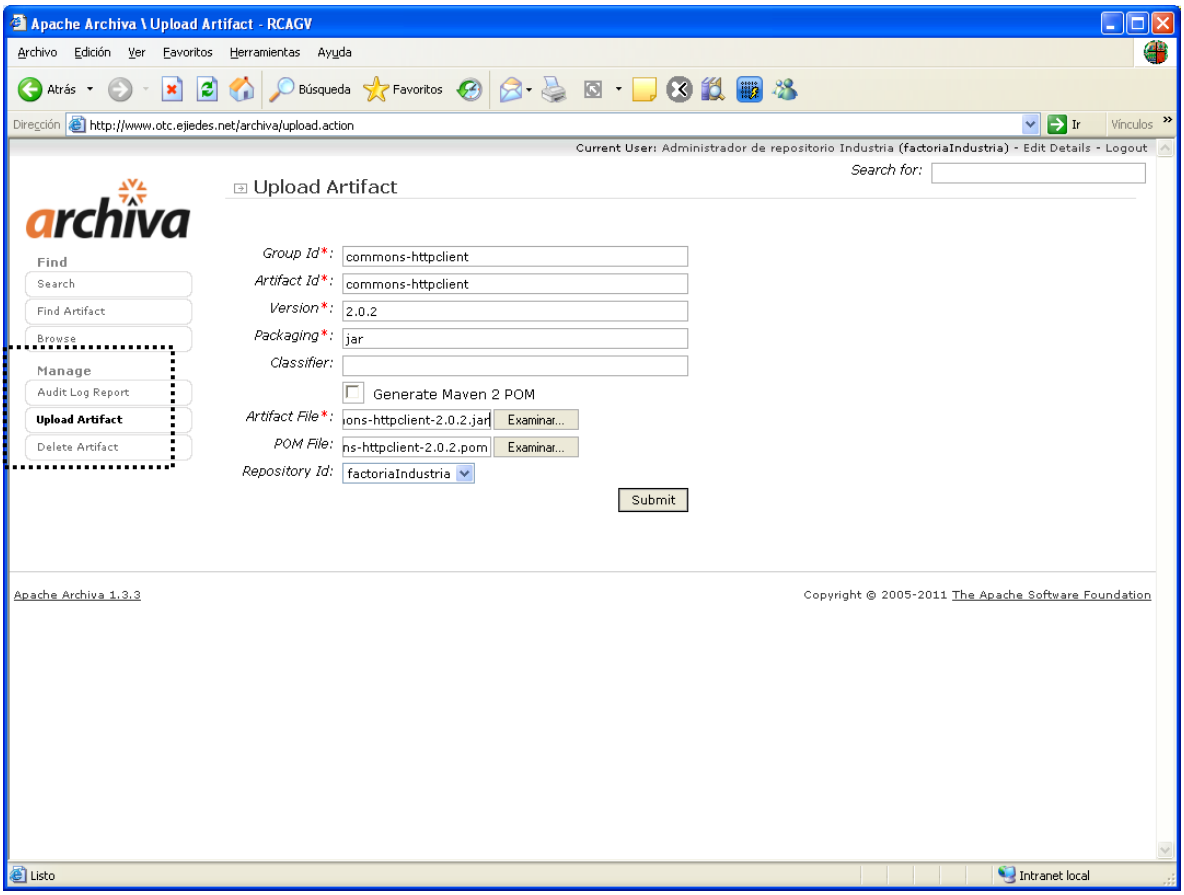
Aquellas factorías de software que dispongan de repositorio privado se les habrá suministrado un usuario y password para el login en Archiva.



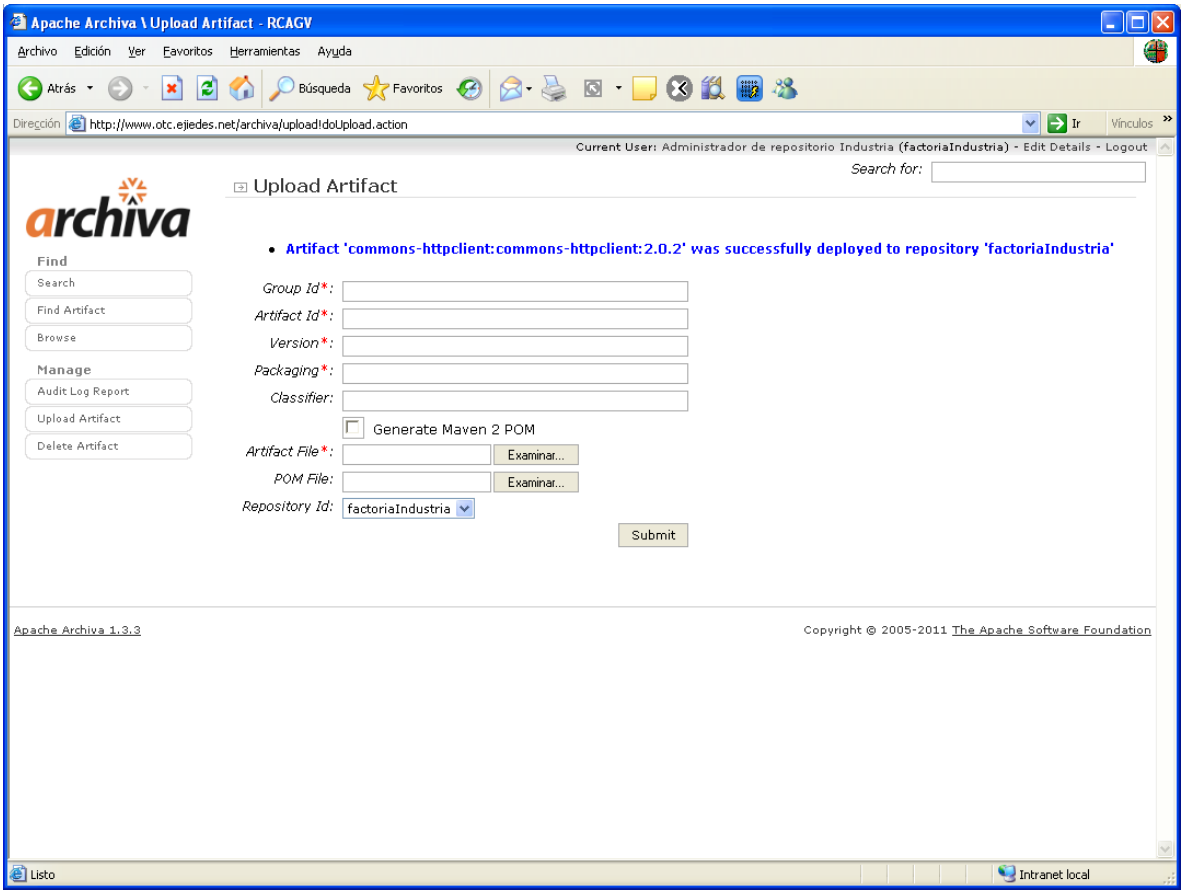
Una vez logados les aparecerán 3 nuevas funcionalidades.

La pantalla de Upload Artifact permitirá que desde la interfaz web se puedan subir artefactos al repositorio privado. Los campos con asterisco son obligatorios. Ésta será la única via para la introducción de artefactos de terceros, es decir, que no pertenezcan a un *Group ID* del tipo *com.ejie.bbb,* puesto que las tareas ant en servidor te controlan que no pueda ser de otro *Group ID*.

En el siguiente ejemplo se muestra como subir la librería de terceros commons-httpclient-2.0.2.jar



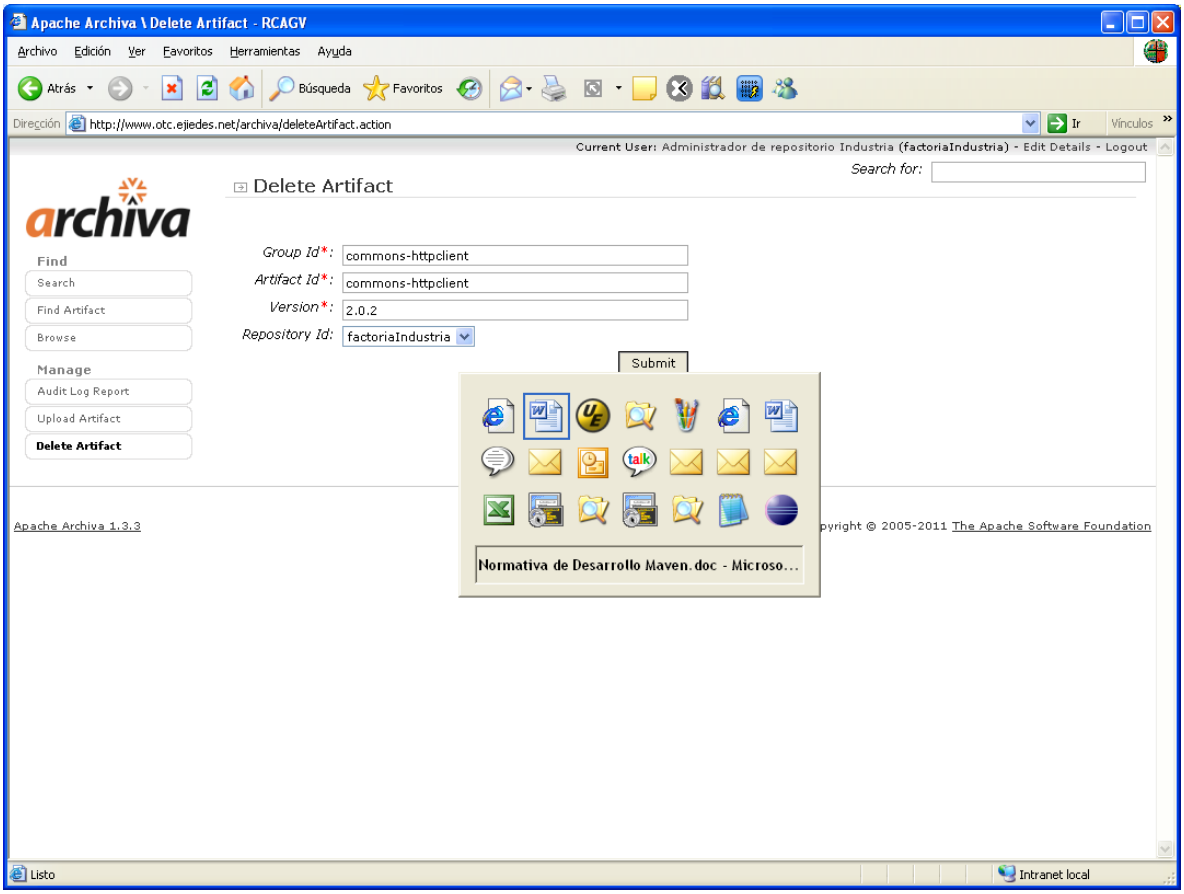
Una vez subido el artefacto dará el siguiente mensaje:



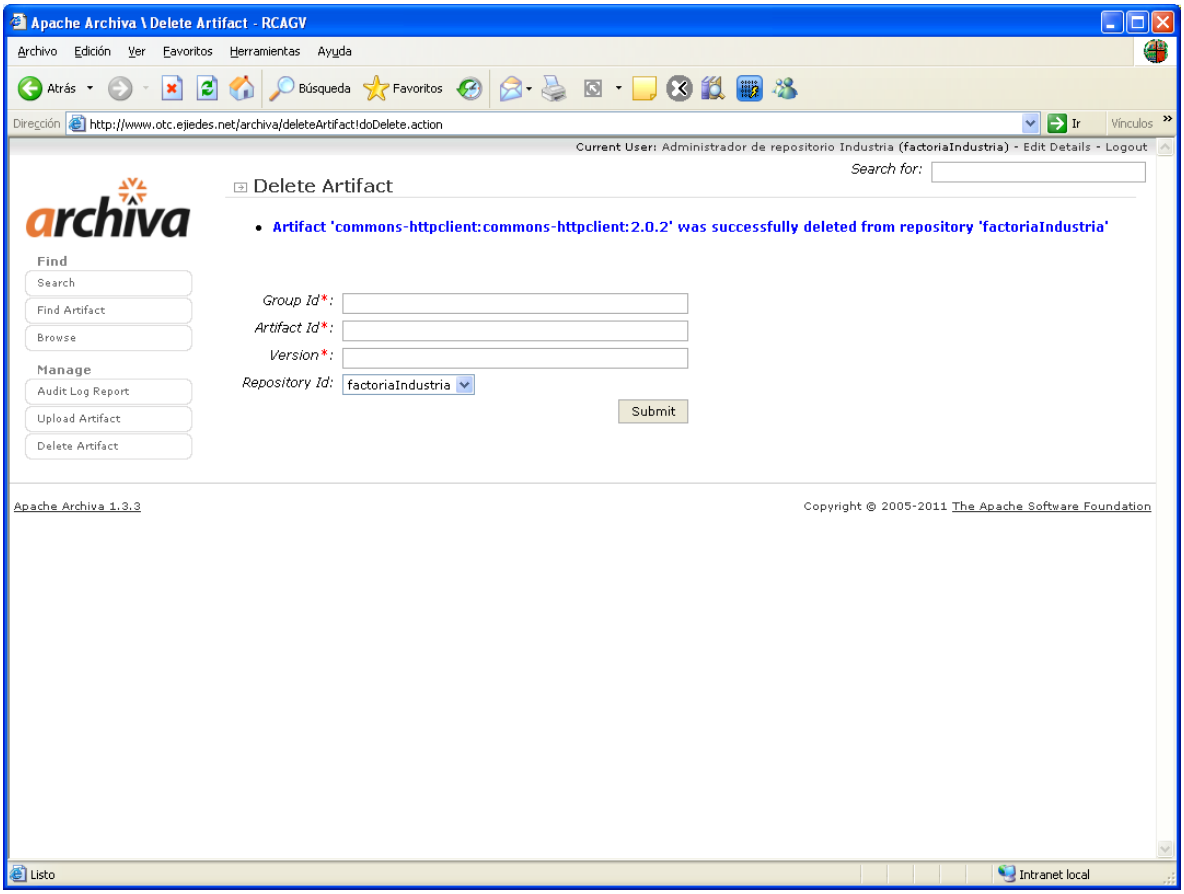
2.7. Borra artefactos (repositorio privado)

Aquellas factorías de software que dispongan de repositorio privado podrán borrar artefactos de su repositorio.

Así siguiendo el ejemplo anterior se podría borrar la librería subida anteriormente de la forma.



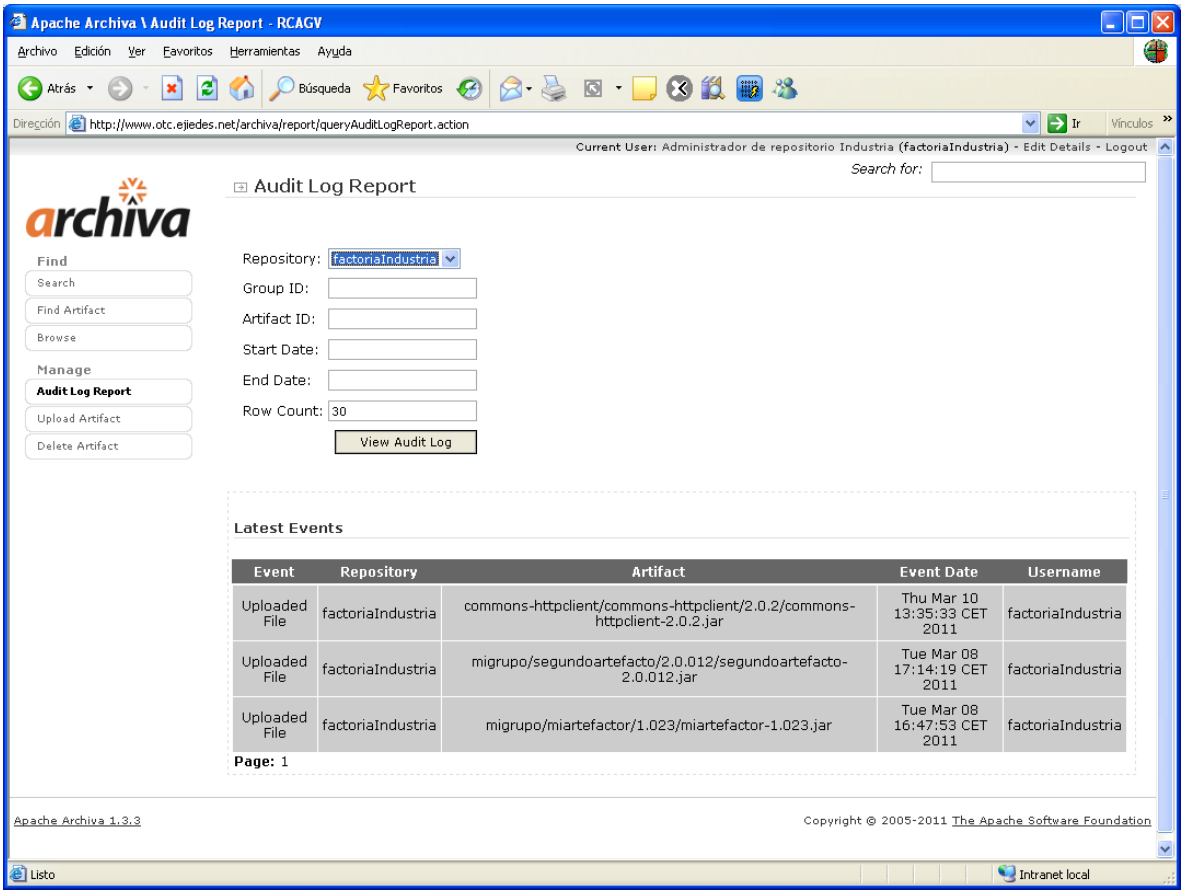
Obtendríamos el siguiente mensaje de confirmación.



2.8. Auditoria (repositorio privado)

Aquellas factorías de software que dispongan de repositorio privado podrán realizar una auditoría de las operaciones realizadas sobre su repositorio.

Permite auditar artefactos concretos y permite filtrar por fechas. La pantalla de auditoría sería la siguiente.



3. Herramientas de desarrollo en equipo local

Los equipos de desarrollo se podrán conectarán a los repositorios de EJIE vía tareas ant para descargar las librerías en su local. Además siempre se podrán conectar a la aplicación de Archiva para navegar en los repositorios y descargar los artefactos manualmente.

3.1. Tareas Ant y normativa ficheros POM

Se tendrá instalado en el equipo local Maven a partir de un rar suministrado por instalaciones con un repositorio en local con el contenido necesario.
Se descomprime apache-maven-2.0.9.rar en c:\ y en el caso de utilizar un servidor Proxy habría que editar el fichero C:\apache-maven-2.0.9\conf\ settings.xml especificando el host, puerto y usuario y comtaseña de la forma siguiente:

```
<settings>
<localRepository>c:\apache-maven-
            2.0.9\repository</localRepository>
  <proxies>
    <proxy>
              <active>true</active>
              <protocol>http</protocol>
              <host>intercon</host>
              <port>8080</port>
              <username>usuario</username>
              <password>password</password>
              <nonProxyHosts>www.otc.ejiedes.net</nonProxyHosts>
</proxy>
</proxies>
 <servers>
  </servers>
  <mirrors>
  </mirrors>
  <profiles>
 </profiles>
</settings>
```

Una vez descomprimido Maven se ha de establecer la variable de entorno M2_HOME a C:\apache-maven-2.0.9.

Una vez instalado maven se creará en cada módulo (proyecto de Eclipse):

- Módulo EAR: pom.xml con las dependencias y un build.xml
- Módulo War: un pom.xml con las dependencias y un build.xml
- Módulo Classes Shlib: un pom.xml con las dependencias y un build.xml

También puede ser necesario instalar “maven-ant-tasks-2.1.3.jar”. Se propone usar la ruta: ${M2_HOME}/maven-ant-tasks/maven-ant-tasks-2.1.3.jar.

3.1.1. Módulo EAR

Se creará un build.xml con el siguiente contenido:

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

<!DOCTYPE project>

<project name="bbbEAR" default="mavenRunDependencies"
       xmlns:artifact="antlib:org.apache.maven.artifact.ant">

      <!-- Permite el uso de variables de entorno -->
       <property environment="env" />

       <target name="mavenRunDependencies" description="Resuelve las
       dependencias del proyecto desde Maven" depends="cleanLib">

         <path id="maven-ant-tasks.classpath"
       path="${env.M2_HOME}/maven-ant-tasks/maven-ant-tasks-2.1.3.jar"
       />

         <typedef resource="org/apache/maven/artifact/ant/antlib.xml"
       uri="antlib:org.apache.maven.artifact.ant" classpathref="maven-
       ant-tasks.classpath" />

         <artifact:dependencies settingsFile="C:\apache-maven-
       2.0.9\conf\settings.xml"/>

         <artifact:mvn pom="pom.xml" mavenHome="C:\apache-maven-2.0.9"
       fork="true">

              <arg value="package"/>

         </artifact:mvn>

       </target>

      <target name="cleanLib" description="Borrar todos los jar. Si se
       borra una dependencia del pom, que se borre.">

           <delete>

                 <fileset dir="./">

                 <include name="**/*.jar"/>

            </fileset>

            </delete>

      </target>

</project>
```

Existirá también un fichero pom.xml con las dependencias como el que sigue:

```
<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.ejie.bbb</groupId>
            <artifactId>bbbEAR</artifactId>
            <packaging>pom</packaging>
            <version>1.0-SNAPSHOT</version>
            <name>bbbEAR</name>
            <url>http://maven.apache.org</url>
            <dependencies>
             . . .
              <dependency>
                    <groupId>org.springframework</groupId>
                    <artifactId>spring-webmvc</artifactId>
                    <version3.0.5.RELEASE</version>
              </dependency>
            . . .

            </dependencies>
            <repositories>
              <repository>
                    <id>ejie</id>
                    <name>Repositorio EJIE</name>
            <url>http://www.otc.ejiedes.net/archiva/repository/repoEJI
            E</url>
                    <snapshots>
                          <enabled>true</enabled>
                    </snapshots>
              </repository>
            </repositories>
            <build>
              <plugins>
                    <plugin>
                          <groupId>org.apache.maven.plugins</groupId>
                          <artifactId>maven-dependency-
            plugin</artifactId>
                          <executions>
                                <execution>
                                      <id>copy-dependencies</id>
                                      <phase>package</phase>
                                      <goals>
                                            <goal>copy-
            dependencies</goal>
                                      </goals>
                                      <configuration>

              <outputDirectory>./EarContent/APP-
            INF/lib</outputDirectory>

              <overWriteReleases>false</overWriteReleases>
```

```
              <overWriteSnapshots>true</overWriteSnapshots>

              <excludeTransitive>false</excludeTransitive>

              <excludeScope>provided</excludeScope>
                                    </configuration>
                                </execution>
                            </executions>
                        </plugin>
              </plugins>
            </build>
</project>
```

Es importante que el elemento del group id sea un com.ejie.bbb en todos los pom. En el caso del EAR no se publicará dicho artefacto.
Como repositorio se especificará el repositorio de EJIE http://www.otc.ejiedes.net/archiva/repository/repoEJIE o uno propio de la factoría de software.
En la sección de plugins se especifica el directorio destino de las librerías. En el caso del EAR será *./EarContent/APP-INF/lib*.

3.1.2. Módulo War

Se creará un build.xml con el siguiente contenido, análogo al del EAR:

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

<!DOCTYPE project>

<project name="bbbModuloWAR" default="mavenRunDependencies"
            xmlns:artifact="antlib:org.apache.maven.artifact.ant">

            <target name="mavenRunDependencies" description="Resuelve
            las dependencias del proyecto desde Maven">

              <path id="maven-ant-tasks.classpath"
            path="${ant.home}/lib/maven-ant-tasks-2.1.1.jar" />

              <typedef
            resource="org/apache/maven/artifact/ant/antlib.xml"
            uri="antlib:org.apache.maven.artifact.ant"
            classpathref="maven-ant-tasks.classpath" />

              <artifact:dependencies settingsFile="C:\apache-maven-
            2.0.9\conf\settings.xml"/>

              <artifact:mvn pom="pom.xml" mavenHome="C:\apache-maven-
            2.0.9" fork="true">

                    <arg value="package"/>
```

```
              </artifact:mvn>

            </target>

</project>
```

Existirá también un fichero pom.xml con las dependencias como el que sigue:

```
<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.ejie.bbb</groupId>
            <artifactId>bbbmoduloWAR</artifactId>
            <packaging>pom</packaging>
            <version>1.0-SNAPSHOT</version>
            <name>bbbModuloWAR</name>
            <url>http://maven.apache.org</url>
            <dependencies>
             . . .
              <dependency>
                    <groupId>org.springframework</groupId>
                    <artifactId>spring-webmvc</artifactId>
                    <version3.0.5.RELEASE</version>
              </dependency>
            . . .

            </dependencies>
            <repositories>
              <repository>
                    <id>ejie</id>
                    <name>Repositorio EJIE</name>
            <url>http://www.otc.ejiedes.net/archiva/repository/repoEJI
            E</url>
                    <snapshots>
                          <enabled>true</enabled>
                    </snapshots>
              </repository>
            </repositories>
            <build>
              <plugins>
                    <plugin>
                          <groupId>org.apache.maven.plugins</groupId>
                          <artifactId>maven-dependency-
            plugin</artifactId>
                          <executions>
                                <execution>
                                      <id>copy-dependencies</id>
                                      <phase>package</phase>
                                      <goals>
                                            <goal>copy-
            dependencies</goal>
                                      </goals>
```

```
                                        <configuration>

              <outputDirectory>./WebContent/WEB-
            INF/lib</outputDirectory>

              <overWriteReleases>false</overWriteReleases>

              <overWriteSnapshots>true</overWriteSnapshots>

              <excludeTransitive>false</excludeTransitive>

              <excludeScope>provided</excludeScope>
                                        </configuration>
                                </execution>
                            </executions>
                        </plugin>
              </plugins>
            </build>
</project>
```

Es importante que el elemento del group id sea un com.ejie.bbb en todos los pom. En el caso del WAR no se publicará dicho artefacto.
Como repositorio se especificará el repositorio de EJIE http://www.otc.ejiedes.net/archiva/repository/repoEJIE o uno propio de la factoría de software.
En la sección de plugins se especifica el directorio destino de las librerías. En el caso del WAR será *./WebrContent/WEB-INF/lib*.

3.1.3. Módulo Classes Shlib

En el caso de que se pretenda crear una librería para ser utilizada por otras aplicaciones se generará un artefacto jar en un proyecto Java de Eclipse de tipo ShLib.
Se creará un build.xml con el siguiente contenido, análogo al del EAR:

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

<!DOCTYPE project>

<project name=" bbbShLibClasses" default="mavenRunDependencies"
            xmlns:artifact="antlib:org.apache.maven.artifact.ant">

            <target name="mavenRunDependencies" description="Resuelve
            las dependencias del proyecto desde Maven">

              <path id="maven-ant-tasks.classpath"
            path="${ant.home}/lib/maven-ant-tasks-2.1.1.jar" />

              <typedef
            resource="org/apache/maven/artifact/ant/antlib.xml"
            uri="antlib:org.apache.maven.artifact.ant"
            classpathref="maven-ant-tasks.classpath" />
```

```
              <artifact:dependencies settingsFile="C:\apache-maven-
            2.0.9\conf\settings.xml"/>

              <artifact:mvn pom="pom.xml" mavenHome="C:\apache-maven-
            2.0.9" fork="true">

                    <arg value="package"/>

              </artifact:mvn>

            </target>

</project>
```

Existirá también un fichero pom.xml con las dependencias como el que sigue:

```
<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.ejie.bbb</groupId>
            <artifactId>bbbShLibClasses</artifactId>
            <packaging>jar</packaging>
            <version>1.0-RELEASE</version>
            <name>bbbShLibClasses</name>
            <url>http://maven.apache.org</url>
            <dependencies>
             . . .
              <dependency>
                    <groupId>org.springframework</groupId>
                    <artifactId>spring-webmvc</artifactId>
                    <version3.0.5.RELEASE</version>
              </dependency>
            . . .

            </dependencies>
            <repositories>
              <repository>
                    <id>ejie</id>
                    <name>Repositorio EJIE</name>
            <url>http://www.otc.ejiedes.net/archiva/repository/repoEJI
            E</url>
                    <snapshots>
                          <enabled>true</enabled>
                    </snapshots>
              </repository>
            </repositories>
            <build>
              <plugins>
                    <plugin>
                          <groupId>org.apache.maven.plugins</groupId>
```

```
                        <artifactId>maven-dependency-
            plugin</artifactId>
                        <executions>
                              <execution>
                                    <id>copy-dependencies</id>
                                    <phase>package</phase>
                                    <goals>
                                          <goal>copy-
            dependencies</goal>
                                    </goals>
                                    <configuration>

              <outputDirectory>./lib</outputDirectory>

              <overWriteReleases>false</overWriteReleases>

              <overWriteSnapshots>true</overWriteSnapshots>

              <excludeTransitive>false</excludeTransitive>

              <excludeScope>provided</excludeScope>
                                    </configuration>
                              </execution>
                        </executions>
                  </plugin>
              </plugins>
            </build>
</project>
```

Es importante que el elemento del *groupid* sea un com.ejie.bbb en todos los pom. A la hora de publicar la librería vía tareas ant se controla que ésto sea así.
En el caso de una librería ShLib se publicará en el repositorio, por lo tanto es importante que se publique con una versión Release las versiones definitivas. El elemento *artifactId* debrá ajustarse a la nomenclatura *bbbShLibClasses* y el *packaging* ser *jar*.
Como repositorio se especificará el repositorio de EJIE
http://www.otc.ejiedes.net/archiva/repository/repoEJIE o uno propio de la factoría de software.
En la sección de plugins se especifica el directorio destino de las librerías. En el caso del jar será *./ /lib*.