

### Introducción.

---

Maven es una herramienta de línea de comandos que nos ayuda con la gestión del proyecto (tareas habituales).

Con comandos simples, o en nuestro caso de manera gráfica con el plugin incluido en eclipse, es capaz de crear una estructura de directorios para nuestro proyecto, para lo que dispone de varias plantillas en función del tipo de proyecto del que se trate, incluso se pueden personalizar. Maven se encarga de generar la estructura del sitio, carpetas para el código fuente del proyecto, la estructura de test, ficheros de configuración y datos, etc.

Aunque el uso fundamental de Maven es la gestión de dependencias, .jar que una librería necesita. A través del archivo pom.xml le se indica a Maven qué dependencias tiene el proyecto (.jar o librerías externas), Maven se encarga de descargar las librerías y todas aquellas dependencias que tuvieran esas librerías e insertarlas en nuestro proyecto. Para ello se conecta a los repositorios Maven (como MavenCentral), aunque es posible instalar un servidor Nexus y generar nuestro propio repositorio.

Todas aquellas librerías que se descargan se almacenan en la carpeta .m2 de la carpeta del usuario.

Sin necesidad prácticamente de configurar nada, Maven sabe cómo borrar los .class, compilar, generar el .jar, generar el javadoc y generar documentación web con varios tipos de informes (métricas, código duplicado, etc.)

Maven se encarga de pasar automáticamente nuestros test de prueba cuando compilamos (escritos con JUnit y que veremos un poco más adelante), Si los test fallan, lanza un mensaje por consola y no compila el proyecto. Incluso, es capaz de generar un .zip de distribución en el que van todos los .jar necesarios y ficheros de configuración de nuestro proyecto.

A pesar de que Maven es una herramienta de línea de comandos, su uso habitual es a través de la interfaz gráfica que emplea Eclipse.

La mejor forma de ver el funcionamiento de Maven es con un ejemplo práctico, vamos a incluir la librería JasperReports que se usa en los ciclos para generar .pdf's.

### Repositorios Maven

---

En <http://www.ibiblio.org/maven2/> se encuentra el repositorio oficial de .jar de Maven. Ahí están los groupId de casi todos los .jar de libre distribución que circulan por internet, como veremos más adelante hay muchos repositorios que podemos usar para obtener las librerías.

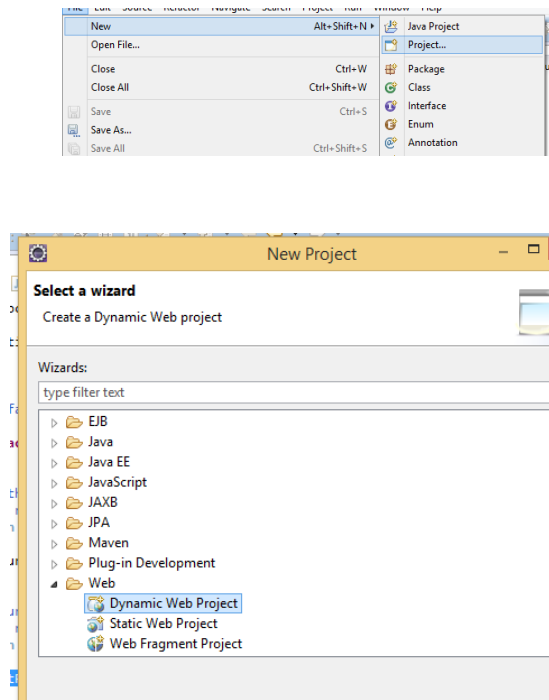
Las librerías que Maven descarga las almacena en un repositorio local, de manera que si las vuelve a necesitar no las descarga de nuevo. Estos repositorios están en:

\$HOME/.m2 en unix/linux

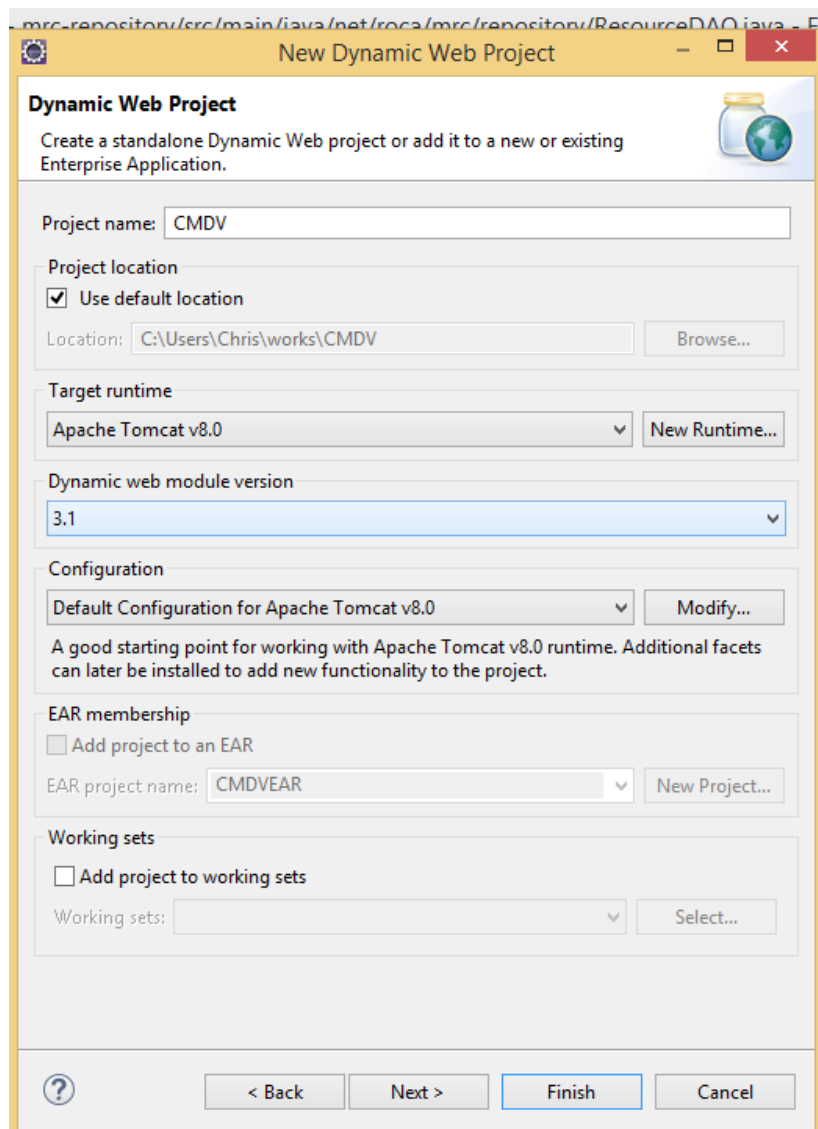
C:\Documents and Settings\usuario\.m2 en Windows

Es posible configurar Maven para que use más repositorios, tanto de internet como otros que montemos internamente en nuestra red local si la tenemos (repositorio Nexus). Nexus posibilita que todos los programadores puedan acceder o subir .jar a ese repositorio y compartirlos.

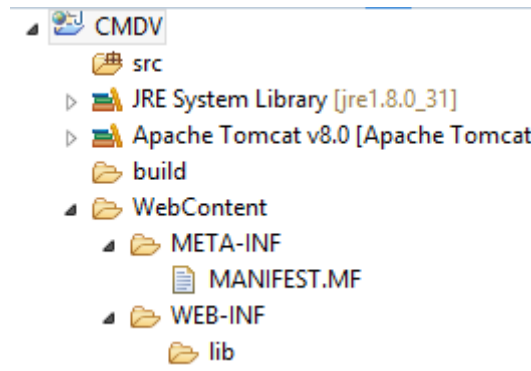
## Crear un proyecto Maven.



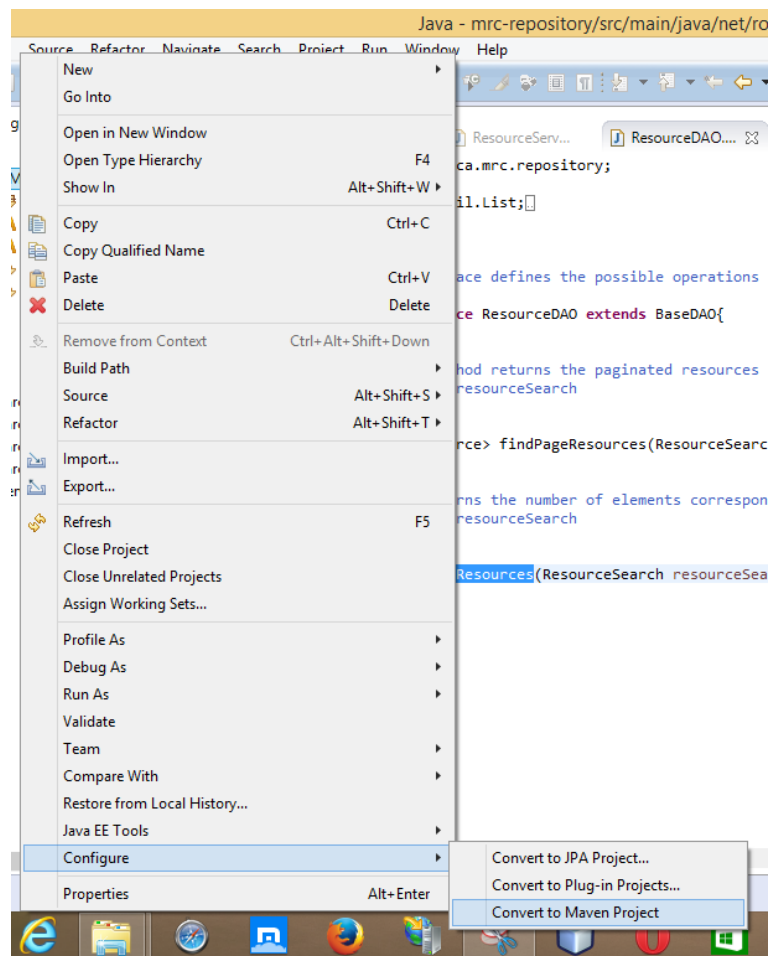
Configuración básica, selección del servidor, Dynamic web module version...

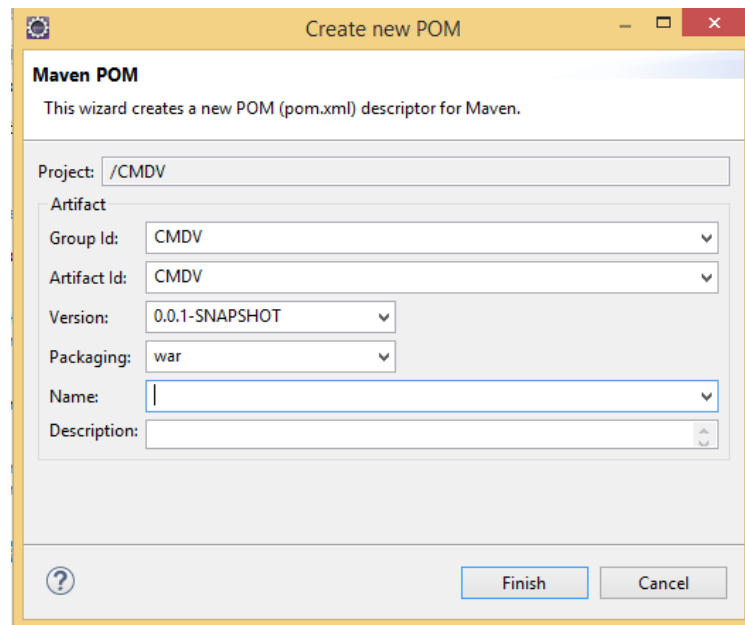


Que da como resultado un proyecto similar a lo mostrado en la siguiente captura:

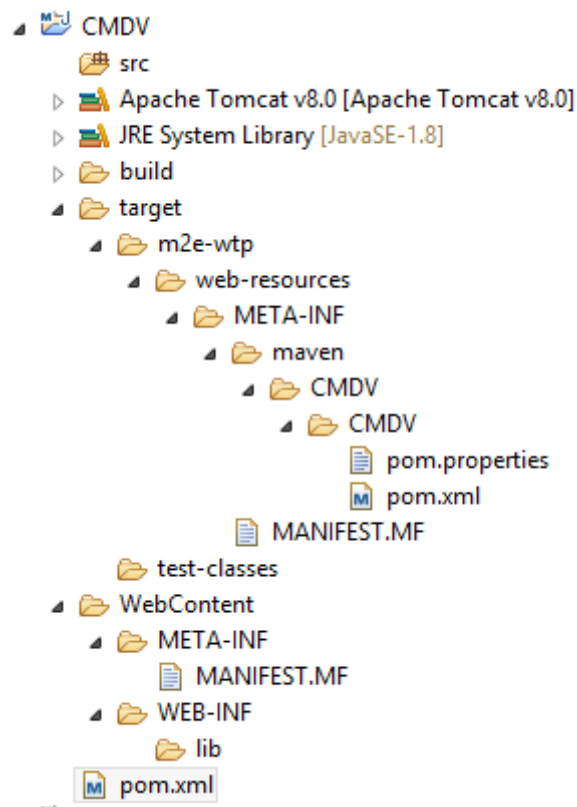


Ahora vamos a “mavenizarlo”





Lo que genera la siguiente estructura:



Donde el archivo más interesante es el pom.xml (archivo de configuración de Maven) y que tiene esta pinta:

```

<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/xsd/maven-
4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>CMDV</groupId>
  <artifactId>CMDV</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <build>
    <sourceDirectory>src</sourceDirectory>
    <plugins>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.1</version>
        <configuration>
          <source>1.8</source>
          <target>1.8</target>
        </configuration>
      </plugin>
      <plugin>
        <artifactId>maven-war-plugin</artifactId>
        <version>2.4</version>
        <configuration>
          <warSourceDirectory>WebContent</warSourceDirectory>
          <failOnMissingWebXml>>false</failOnMissingWebXml>
        </configuration>
      </plugin>
    </plugins>
  </build>
  <organization>
    <name>Cristóbal Martínez (CMDV)</name>
  </organization>
</project>

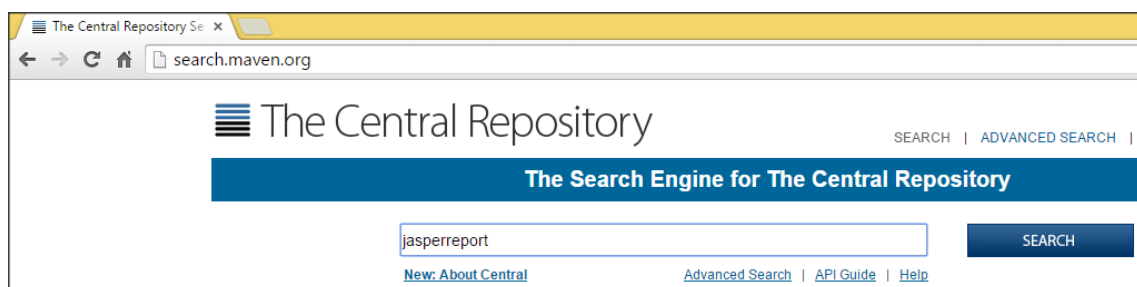
```

Carpeta y subcarpeta dentro de .m2 donde Maven genera los compilados, el war de la aplicación...

Para ver un ejemplo sencillo del funcionamiento de Maven, vamos a incluir la librería de jasperreports en el proyecto, para ver como Maven descarga la librería, sus dependencias y cómo organiza los .jar.

Prácticamente la totalidad de librerías opensource ofrecen el código Maven junto a los .jar para su descarga.

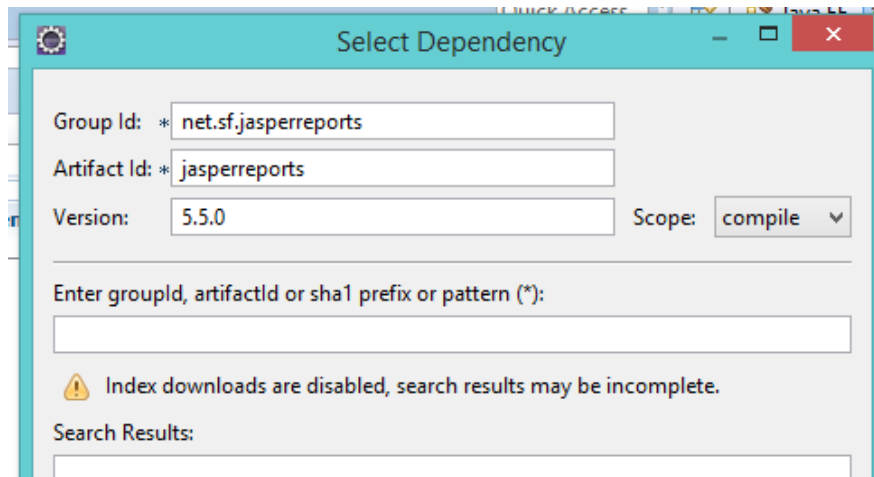
Aquí empieza la magia de Maven. Vamos a un repositorio Maven cualquiera, como ya he comentado uno de los más conocidos es MavenCentral (aunque hay otros como <http://mvnrepository.com/> o <https://maven-repository.com> por poner dos ejemplos ) y buscamos la librería.



En cualquier caso llegamos a algo similar a esto:



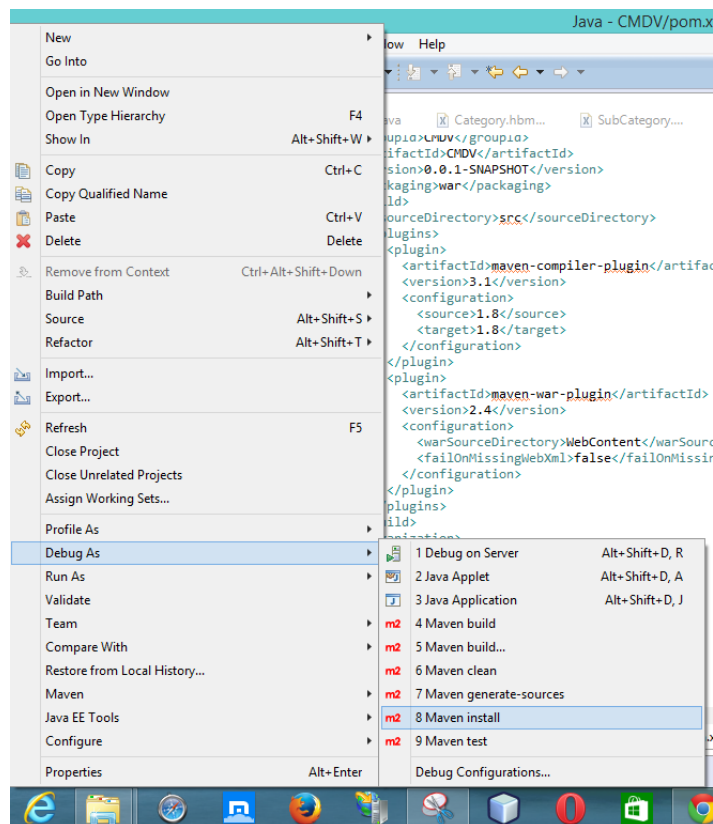
Podemos copiar y pegar el código en el .pom, o usar el asistente de eclipse para generar dependencias.



En cualquiera de los casos el .pom queda así:

```
29     <name>Cristobal Martinez (CMDV)</name>
30 </organization>
31 <dependencies>
32   <dependency>
33     <groupId>net.sf.jasperreports</groupId>
34     <artifactId>jasperreports</artifactId>
35     <version>5.5.0</version>
36   </dependency>
37 </dependencies>
```

Ahora viene la magia de Maven.



Ejecuta las acciones de Maven, builds, descargas...

Lo que genera la típica salida por consola de Maven cuando no hay errores.

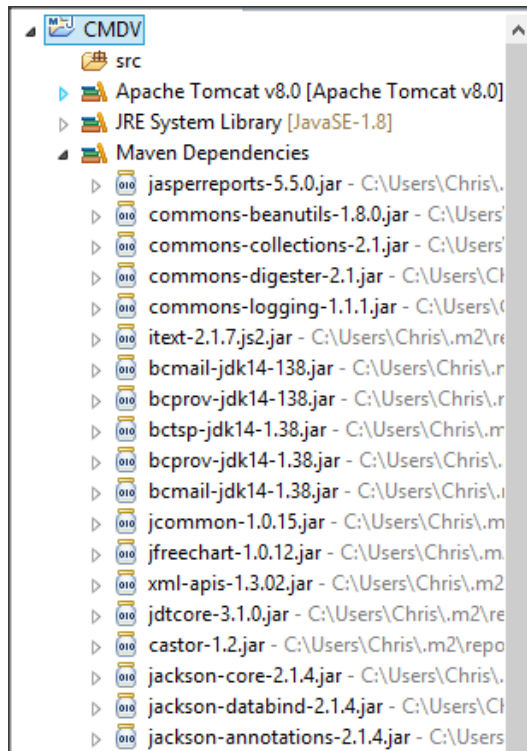
```

[INFO] --- maven-install-plugin:2.4:install (default-install) @ CMDV ---
[INFO] Installing C:\Users\Chris\works\CMDV\target\CMDV-0.0.1-SNAPSHOT.war to C:\Users\Chris\.m2\repository\CMDV\CMDV\0.0.1-SNAPSHOT\CMDV-0.0.1-SNAPSHOT.war
[INFO] Installing C:\Users\Chris\works\CMDV\pom.xml to C:\Users\Chris\.m2\repository\CMDV\CMDV\0.0.1-SNAPSHOT\CMDV-0.0.1-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.699 s
[INFO] Finished at: 2015-03-30T20:33:46+01:00
[INFO] Final Memory: 11M/125M
[INFO] -----

```

Veamos que ha hecho en el proyecto:

Se ha generado una nueva carpeta en el proyecto Maven Dependencies,



Donde podemos encontrar no sólo la librería de jasperreports, sino todas las librerías necesarias para que ésta funcione, incluso las dependencias de las dependencias, todo preparado para poder usarlo en el proyecto. Y de todo se ha encargado Maven de manera automática simplemente incluyendo la dependencia en el archivo pom.

Para configurar todas la librerías necesarias simplemente hay que incluirlas dentro de un tag <dependency>, es más, la versión puede ser una variable y con cambiar el valor de la variable y realizar otro Maven Install el proyecto actualizaría las librerías a la versión deseada.

Veamos que significa la dependencia:

```

<dependency>
  <groupId>net.sf.jasperreports</groupId> (ruta en .m2)
  <artifactId>jasperreports</artifactId> (carpeta que contiene el .jar)
  <version>5.5.0</version> (version de la librería)
</dependency>





```

Este equipo > Disco local (C:) > Usuarios > Chris > .m2 > repository > net > sf > jasperreports > jasperreports > 5.5.0

Nombre	Fecha de modifica...	Tipo	Tamaño
_remote.repositories	30/03/2015 20:37	Archivo REPOSIT...	1 KB
jasperreports-5.5.0.jar	30/03/2015 20:37	Executable Jar File	4.554 KB
jasperreports-5.5.0.jar.md5	30/03/2015 20:37	Archivo MD5	1 KB
jasperreports-5.5.0.jar.sha1	30/03/2015 20:37	Archivo SHA1	1 KB
jasperreports-5.5.0.pom	30/03/2015 20:36	Archivo POM	14 KB
jasperreports-5.5.0.pom.md5	30/03/2015 20:36	Archivo MD5	1 KB
jasperreports-5.5.0.pom.sha1	30/03/2015 20:36	Archivo SHA1	1 KB

Además Maven genera una carpeta para el proyecto, donde coloca los compilados el .war...

Este equipo ▸ Disco local (C:) ▸ Usuarios ▸ Chris ▸ .m2 ▸ repository ▸ CMDV ▸ CMDV ▸ 0.0.1-SNAPSHOT

^	Nombre	Fecha de modifica...	Tipo	Tamaño
	 _remote.repositories	30/03/2015 20:38	Archivo REPOSIT...	1 KB
	 CMDV-0.0.1-SNAPSHOT.pom	30/03/2015 20:36	Archivo POM	2 KB
	 CMDV-0.0.1-SNAPSHOT.war	30/03/2015 20:38	Archivo WAR	15.449 KB
	 maven-metadata-local.xml	30/03/2015 20:38	Archivo XML	1 KB