# FaMa



# **Apache Maven**











# Contenido

Guía rápida de Maven en FaMa Aspectos previos.	4 5
Configuración	
I letting at 4 m	





## Guía rápida de Maven en FaMa.

Maven se basa en el concepto de build lifecycle. Un build lifecycle (o ciclo de vida de construcción de un proyecto) está compuesto de diversas fases, y asociado a un proyectos. Por defecto, existen varios ciclos de vida, aunque el que utilizaremos fundamentalmente para FaMa será el default (y puede también que clean). Es decir, un ciclo de vida de un proyecto Maven tiene varias fases, ejecutándose en cada una de ellas unas acciones determinadas (goals). Dichas fases tienen por defecto unas acciones predeterminadas asociadas, pudiéndose asociar nuevas acciones a cada una de las fases.

Cabe destacar que dichas fases se ejecutan en orden secuencial, y que si ejecutamos una fase, se ejecutarán ordenadamente todas las anteriores hasta llegar a la actual, que por supuesto también será ejecutada.

Para FaMa, las fases del ciclo de vida default más usuales en la ejecución serán:

- compile: la utilizaremos para resolver las dependencias del proyecto y compilarlo.
- install: en esta fase se crea el .jar y se envía al repositorio local (en la propia máquina donde se ejecuta).
- deploy: finalmente, en esta fase subimos el .jar al repositorio Maven.

En general, este será el orden en que llamemos a las fases tras hacer un checkout. Primero resolveremos dependencias y compilaremos el proyecto para comprobar que todo funciona correctamente con la fase compile.

Una vez realicemos los cambios que debamos, llamaremos a install (lo que llevará los cambios al repositorio local).

Para la mayoría de los proyectos que dependan del proyecto actual, con las dependencias se actualizarán automáticamente (pues tirar de los .jar del repositorio local). Sin embargo, los proyectos usuarios de FaMa (como el benchmark o el test suite) que necesitan acceder a algunos .jar con el jarjarclassloader, deberán tener los ficheros en local (teniendo que ejecutar sobre ellos la fase compile explicada previamente).

Finalmente, cuando estemos seguros de los cambios realizados y hayamos hecho las pruebas pertinentes, ejecutaremos la fase deploy para almacenar el .jar definitivo en el repositorio.

Otros goals o fases útiles pueden ser:

- dependency:purge-local-repository: (goal) actualiza el repositorio local, descargando desde el remoto las versiones de las dependencias correspondientes del proyecto.
- package: (fase) genera el .jar correspondiente al proyecto, aunque sin instalarlo en el repositorio local (por defecto aparecerá en la carpeta target).

#### FaMa & Maven





Por supuesto, aparte de esto será necesario ir subiendo el código fuente con el cliente de subversión correspondiente.

Para un análisis detallado, consultar http://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html

Respecto a los parámetros de entrada que necesita recibir Maven, todos aquellos que no son tomados por línea de comandos los recibe a través del POM.xml (el archivo de configuración básico de un proyecto Maven). En él, establecemos dependencias entre proyectos (para más información sobre las dependencias, consultar <a href="http://maven.apache.org/guides/introduction/introduction-to-dependency-mechanism.html">http://maven.apache.org/guides/introduction/introduction-to-dependency-mechanism.html</a> ), asociamos nuevas acciones a cada fase (consultar los diversos plugins de Maven en <a href="http://maven.apache.org/plugins/index.html">http://maven.apache.org/plugins/index.html</a> ), o configuramos las fases ya existentes, entre otros.

Para una descripción en profundidad sobre el POM, consultar http://maven.apache.org/pom.html .

Por otro lado, para información más relativa al usuario, o para un conjunto de proyectos, tenemos el archivo settings.xml (<a href="http://maven.apache.org/settings.html">http://maven.apache.org/settings.html</a>)

## Aspectos previos.

En primer lugar, hay que destacar dos aspectos importantes:

- Para utilizar Maven, tenemos dos opciones:
  - Usar la versión embebida de Maven que trae por defecto el plugin de Maven para eclipse m2eclipse.
  - Usar una instalación externa de Maven.

La recomendación es utilizar una instalación externa de Maven, para no estar obligados a utilizar éste siempre desde eclipse (y no poder integrarlo en el futuro con otras aplicaciones). En caso de querer utilizar la versión embebida, podremos saltarnos el paso 1 de la instalación

 Por otro lado, ahora que Maven ha sido implantando en FaMa, es muy recomendable hacer uso de los proyectos siempre con Maven, aunque es posible seguir utilizando los proyectos al margen de éste.





#### Instalación

- Instalar el plugin de eclipse m2eclipse, tal como se detalla en <a href="http://m2eclipse.codehaus.org/">http://m2eclipse.codehaus.org/</a>. En caso de problemas con la instalación, no incluir en ella el plugin de pom en estado de incubation.
- 3. Ejecutar eclipse desde el jre perteneciente al jdk. Para ello, hemos de añadir dicho jre a eclipse si no lo tenemos hecho (Preferences/Java/Installed JREs), e iniciar eclipse desde dicho jre, añadiendo el parámetro -vm a eclipse.ini (situado en el raíz de eclipse. Para dudas, consultar <a href="http://help.eclipse.org/help33/index.jsp?topic=/org.eclipse.platform.doc.user/tasks/running\_eclipse.htm">http://help.eclipse.org/help33/index.jsp?topic=/org.eclipse.platform.doc.user/tasks/running\_eclipse.htm</a>)
- 4. Instalar el certificado necesario para poder trabajar a través de Maven con el repositorio de la forja de rediris.
  El certificado se encuentra en <a href="http://secure.globalsign.net/cacert/educational.crt">http://secure.globalsign.net/cacert/educational.crt</a>
  Una vez lo tengamos, hemos de añadirlo al jdk de java, haciendo uso de la herramienta "keytool" del jdk (se encuentra en "bin/keytool", suponiendo que estamos situados en directorio raiz del jdk). El comando a escribir es el siguiente.

keytool -keystore \$JAVA\_HOME/jre/lib/security/cacerts -import -file /ruta/al/certificado/

El certificado ha de estar en nuestro sistema de archivos (en local). Hecho esto, nos preguntara la contraseña, la cual es "changeit" (sin comillas).

En caso de cambiar o reinstalar el jdk, será necesario repetir el proceso.

# Configuración.

Ahora que ya tenemos instalado Maven, el plugin para eclipse, y el certificado, podemos configurar algunos detalles del entorno de trabajo.

Como antes se comentó, ha cambiado en parte la estructura de directorios de los proyectos de FaMa. Los .class aparecerán ahora en una carpeta llamada target, las librerías internas de FaMa (questions, core, reasoners...) aparecerán en algunos casos dentro de una carpeta llamada lib\_noscm (y en otras directamente no aparecerán en ninguna carpeta del proyecto, pero sí estarán referenciadas), y las librerías externas (como xerces, choco-solver...) estarán en resources/lib. Por otro lado, el config.xml estará en resources.

www.isa.us.es/fama

#### FaMa & Maven





Para entender el por qué de estos cambios, sería conveniente primero entender el funcionamiento de Maven. Para ello, acudir a los apartados "Guiá rápida de Maven en FaMa" y "Referencias".

A continuación, se relatan una serie de pasos para terminar de configurar Maven y eclipse.

- 1. Ignorar a nivel de SVN las carpetas lib\_noscm y target (no queremos almacenar en el repositorio ni los .class, ni las librerías internas para cada proyecto, de esta es de una de las cosas que se ocupa Maven).Para ello, tenemos dos opciones:
  - Por cada proyecto, seleccionar las carpeta, y añadir a svn:ignore (click secundario, Team/ svn:ignore).
  - En eclipse, Preferences/Team/Ignored Resources, y añadir ahí los patrones correspondientes (target y \*\_noscm, por ejemplo).
- Escribir en el settings.xml los datos correspondientes al repositorio de la forja.
   Para cualquier información relacionada con settings.xml, consultar
   <a href="http://maven.apache.org/settings.html">http://maven.apache.org/settings.html</a>
   En nuestro caso, deberemos dirigirnos a la sección Servers. El "id" será forja, y con añadir los campos "username" y "password", con sus correspondientes valores, será suficente.

#### Utilización.

Una vez hecho lo anterior, podremos comenzar a hacer checkout de los proyectos Maven de FaMa. Es recomendable hacer checkout (nada de checkout as a Maven Project), pues ya están perfectamente configurados para funcionar correctamente.

Para ejecutar un "goal", seleccionamos el proyecto, y nos vamos a Run as. Existen algunos predefinidos, como install, package o test. En caso de no encontrar el que buscamos, elegimos "Build...", y en goals escribimos las fases o goals que queramos ejecutar.

Cabe destacar que las primeras veces que ejecutemos una determinada tarea o fase en Maven, éste tendrá que descargarse la última versión de cada uno de los plugins que ofrecen la funcionalidad de dichas fases o tareas, por lo que es probable que las primeras ejecuciones de Maven tarden más de lo normal.

www.isa.us.es/fama