

Instrucciones de instalación Eclipse

1. Help - > Eclipse Market Place
 - a. Buscamos m2eclipse e instalamos
 - i. Maven Integration for Eclipse
 - ii. Maven Integration for Eclipse(WTP)
2. Copiar el archivo setting.xml desde /etc/maven a la .m2 (del directorio personal)
3. Windows -> Preferences -> Maven -> Installations -> add: la ruta donde se encuentre maven (varía entre sistemas)
4. Modificar el archivo setting.xml y añadir dentro de las etiquetas de mirror:

```
<mirror>
  <!-- United Kingdom -->
  <id>uk.maven.org</id>
  <url>http://uk.maven.org/maven2</url>
  <mirrorOf>central</mirrorOf>
</mirror>
```
5. Crear un nuevo proyecto (Other/Maven Project) Seleccionamos Next - > maven-archetype-quickstart Next -> Group id : lambdasoft y Artefact id: crysport y seleccionamos Finish.
6. No vamos al proyecto creado y botón derecho RunAS -> Maven Clean - > RunAs -> Run configurations -> maven build -> crysport -> Goals: clean, package site y Apply y Finish. Esperar a que se descarguen las dependencias y actualizaciones.

Maven

-Este resumen respecta al video de Maven que hay en internet-

Cuando generamos un proyecto en maven su descripción viene especificada en un archivo .xml llamado pom (Project Object Model). Si no queremos especificar toda la descripción a mano en el archivo .xml hacemos uso de arquetipos que no son más que plantillas preexistentes de proyectos Maven.

Cuando vamos a crear un proyecto Maven nos pide los siguientes datos:

- Group id: sería como el paquete de nuestro proyecto (ej. com.crysport)
- Artefact id: es el identificador único de nuestro proyecto.
- version: ponemos la versión que decidamos pero normalmente es la 1.0.

Tras esto se nos genera un proyecto que estará incluido en una carpeta con el mismo nombre. El contenido de las carpetas es la siguiente:

- un archivo pom.xml
- Carpeta src/main/java/crysport que contendrá la clase App.java (por defecto contiene un programa de "Hello World!")

- Carpeta src/test/java/cryspor tenemos una clase AppTest con una pruebas que por defecto no hacen nada.
- En la carpeta target encontramos los binario generados de nuestro proyectos maven.

El tener estas clases no es más que una ayuda que proporciona maven para saber nosotros donde poner nuestros fichero de aplicación.

Cabe destacar el formato y contenido del fichero pom.xml:

- packaging: define el tipo de proyecto que se va a generar. Esto es necesario ya que todos los proyectos maven generan un binario
- url: se pone la url de la organización. Nosotros podemos poner la de google sites.
- dependencies: un aspecto a destacar de maven es la forma en la que se gestionan las dependencias. Al crear un proyecto en java se suelen usar librerías, en este aspecto maven nos da soporte. Para definir una dependencia (respecto a una librería en maven) hacemos lo siguiente:
 - Creamos una dependency (dentro del fichero pom.xml) en el que definimos
 - group id, artefact id y la versión usada.
 - scope define en que nivel se necesita la librería, por ejemplo podríamos necesitar una librería solo en el nivel de test.

Quando nosotros compilamos nuestra aplicación Maven descarga la librería del repositorio central y la pone en nuestro repositorio local.

- modules: indica que módulos forman parte de nuestro proyecto. Un módulo puede ser otro proyecto.

Algunas acciones que podemos realizar

- Si lo deseamos podemos decirle a maven que solo nos compile las pruebas (sin generar el .jar) para poder ejecutarlas.
- Si deseamos instalar nuestra aplicación esto lo realizará maven compilando el proyecto e instalando nuestra aplicación en la carpeta personal dentro de la carpeta oculta .m2. Dentro de .m2 también estarán los repositorios que se nos descarguen.

Una de las características más potentes de maven es que soporta proyectos compuestos por varios módulos.

Podemos crear varios tipos de proyectos como:

- Proyectos POM que no generan un binario si no que generan un descriptor general en un archivo .xml.
- Además a partir de un proyecto podemos generar varios proyectos hijos y proyectos que dependan de otros proyectos.