MAVEN

<https://www.javaworld.com/article/2071772/java-app-dev/the-maven-2-pom-demystified.html>

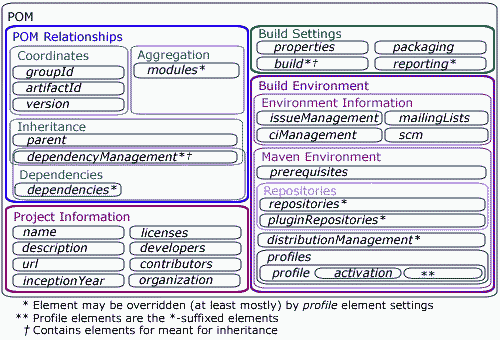
Ver también el ciclo de Maven.

Más que una herramienta de construcción de proyectos, Maven es un *framework* de construcción de proyectos que separa limpiamente el código de los ficheros de configuración, la documentación y las dependencias. De hecho, el proyecto “mora” dentro de la estructura de Maven.

El fichero de configuración de un proyecto dentro de Maven es el .pom (Project Object Model).

Todas las configuraciones del proyecto dentro del .pom pueden agruparse en cuatro unidades lógicas:

* Las relaciones
* La información del proyecto
* La configuración de la construcción (build)
* El entorno de construcción o compilación



**Relaciones**

Hay cuatro tipos de relaciones en Maven: Coordenadas, dependencias basadas en coordenadas, herencia entre proyectos y agregación.

* Coordenadas (coordinates): cada proyecto Maven contiene su identificador único, llamado también las “coordenadas” del proyecto, que actúa como la “dirección” única de un artefacto. Estas coordenadas son <groupId>, <artifactId> y <version>, que son campos obligatorios.
* Dependencias: la gestión de las dependencias es uno de los fuertes de Maven, así como las dependencias transitivas de Maven 2. Con un repositorio local común a todas las dependencias y proyectos, Maven resuelve el problema atroz y recurrente del “Jarmageddon” y el “jar hell”.
* Herencia: un proyecto Maven puede heredar de un padre (definido con el tipo de empaquetado = pom; ver “packaging”). Así, el hijo tendrá que usar las coordenadas del padre entre las etiquetas <parent>, heredando su <groupId> y su <version> y definiendo sólo su <artifactId>. En realidad, todos los .pom heredan de un padre, el “super pom”, lo hagan o no explícitamente.

Es importante resaltar la etiqueta <dependencyManagement>, que es un instrumento muy útil para configurar un conjunto de dependencias a través de un padre común. Ver la etiqueta en detalle.

* Agregación: etiqueta <modules>. Un proyecto que define estos módulos se conoce como “multi-módulo”. Los módulos son proyectos que se ejecutan como un conjunto (“the reactor” los ejecuta en un orden determinado y necesario). Ver la diferencia con una dependencia y su necesidad.

**Información del proyecto:**

Los elementos de esta parte tienen un propósito informativo para quien usa el proyecto.

**Configuración de la construcción (build settings):**

Esta sección se compone de cuatro etiquetas: <build>, <reporting>, <packaging> y <properties>, de las cuales las dos primeras son las más importantes. Empezaremos por las menos:

* Packaging: si no se especifica toma el valor “jar”. Los otros valores posibles son “pom”, “maven-plugin”, “ejb”, “war”, “ear”, “rar”, “par” y “ejb3”. Cada uno de ellos está asociado con una lista de “goals” a ejecutar correspondiente a su ciclo de construcción particular.
* Properties: es un apartado en que se definirán propiedades con sus valores. Dado que hay muchas otras maneras de definir propiedades, no es una característica muy importante.
* Build: define toda la información necesaria para construir / compilar el proyecto, desde la localización del código fuente a la configuración de los plugins de maven que usará, entre otros. Mucha de esta información se herada del .pom padre. Todas las rutas que no contengan “/” serán relativas a ${baseDir} (¿?)