

# Maven

## ¿Qué nos resuelve?

- [Creación automática de los proyectos para poder importarlos desde un IDE.](#)
- [Automatización de tareas de compilación/ejecución de test/cobertura, etc.](#)
- [Manejo de las dependencias de nuestro proyecto.](#)

## ¿Cómo se instala?

<https://maven.apache.org/install.html>

### **Importante: ¿Cómo sé si instale correctamente Maven?**

En la terminal ([linux](#) / [windows](#)) ejecuto el comando  
***mvn --version***

Debería mostrar algo similar a lo siguiente:

***Apache Maven 3.5.0***

***Maven home: /usr/share/maven***

***.....***

## ¿Cómo funciona?

En el directorio raíz del proyecto se deberá crear un archivo con el nombre ***pom.xml***

En su versión más básica se parecerá a esto:

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>nombre-proyecto</groupId>
  <artifactId>nombre-proyecto</artifactId>
  <version>1</version>
</project>
```

Vamos a ir viendo cómo este archivo va a ir creciendo y agregando más información.

## ¿Cómo uso Maven para importar mi proyecto desde Eclipse?

Primero hay que ejecutar los siguientes 2 comandos en la consola:

***mvn eclipse:clean***

Sirve para borrar cualquier configuración del proyecto anterior.

***mvn eclipse:eclipse***

Crea la configuración del proyecto.

***mvn eclipse:eclipse -DdownloadSources=true -DdownloadJavadocs=true***

Crea la configuración del proyecto y baja el código fuente de las bibliotecas utilizadas.

Haciendo estos pasos desde Eclipse, ya vas a poder importarlo como un proyecto cualquiera desde la opción **File -> Import -> Import existing into Workspace** y navegás hasta llegar a la carpeta de tu proyecto.

#### **- Me gustaría usar Java 8 ¿Tengo que hacer algo en particular?**

- Si, en el *pom.xml* vas a tener que decirle que tiene que compilar con la versión 8.

Un ejemplo de cómo quedaría el primer pom que vimos agregándole la opción de Java 8 sería:

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>proyecto</groupId>
  <artifactId>proyecto</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <build>
    <plugins>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.1</version>
        <configuration>
          <source>1.8</source>
          <target>1.8</target>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

Y con esto ya tenemos configurado nuestro proyecto para Java 8.

## ¿Cómo construyo mi proyecto (compilación, test, etc)?

Existen muchos comandos, te voy a enseñar los más comunes:

### ***mvn clean***

Borra todos los archivos compilados que existan.

### ***mvn compile***

Compila todo el código fuente.

### ***mvn test***

Primero compila todo el código fuente si no estuviera compilado y luego ejecuta los tests.

### ***mvn install***

Ejecuta todos los pasos anteriores (***compile***, ***test***, ***etc***) y además junta el código compilado

en un archivo .jar y deja la biblioteca disponible en mi sistema para que la pueda utilizar algún otro proyecto.

## ¿Cómo agrego una dependencia?

### - Antes de saber cómo agregar... ¿Qué es una dependencia?

Podríamos decir que una dependencia en Maven es una biblioteca hecha por terceros que voy a tener disponible para utilizar en mi proyecto.

Si la biblioteca de terceros utiliza otras bibliotecas, entonces Maven se va a dar cuenta de eso y va a bajar esas dependencias también.

- Por lo que voy entendiendo.. ¿JUnit sería una dependencia?
- Síí!!! Y podés agregarlo a través de Maven. Te muestro un ejemplo:

```
<project>
.....
<!-- Dependencias -->
<dependencies>

    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.12</version>
    </dependency>

</dependencies>
.....
</project>
```

Una cosa, si ya habías importado el proyecto en Eclipse, acordate de hacer ***mvn eclipse:eclipse*** y refrescar el proyecto para que tome los cambios.

## Encontré una biblioteca y quiero agregarla a mi proyecto ¿cómo sé que Maven la va a poder bajar?

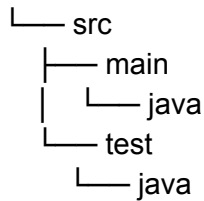
Bueno, Maven baja todas las bibliotecas de un repositorio mundial que se lo conoce como Maven Repository. Entonces entrá al sitio de este repositorio y revisá si está en el repositorio y si está mira incluso el snippet que te muestra cómo agregarla.

Este es el link al repositorio: <http://mvnrepository.com/>

## ¿Dónde agrego las clases en Eclipse?

Primero vas a tener que crear el siguiente árbol de directorios:

```
.
├── pom.xml
```



- En **src/main/java**, van las clases de dominio
- En **src/test/java**, van las clases que van a testear el dominio

Ejecutás nuevamente **mvn eclipse:eclipse**, una vez hecho esto vas a Eclipse, refrescás el proyecto y ya podés empezar agregar las clases :)

## ¿Algo más para saber?

Sí, si usás GIT, acordate de agregar al .gitignore los archivos que se autogeneran (.classpath, .project, target, etc).

## ¿Y si tengo un proyecto existente y quiero que utilice Maven?

Fácil:

- Creás tu archivo pom.xml en el raíz del proyecto
- Borrás todos los archivos de configuración de Eclipse (.project, etc..)
- Creás la estructura de directorios antes especificada y ponés las clases donde corresponden.
- Ejecutás **mvn eclipse:eclipse**
- Listo, ya se puede importar desde Eclipse :D

## ¿Qué links me recomendás para seguir leyendo?

- <https://maven.apache.org/>
- <https://sites.google.com/site/programacionui/temario/01-intro-UI/maven>

## ¿Tenés algún video de ejemplo?

Si, te dejo el link: [https://youtu.be/zJleYRjxM\\_w](https://youtu.be/zJleYRjxM_w)