

# Tutorial

---

## Configuración de Maven

---

## 1. Introducción

---

Maven es un gestor de dependencias para proyectos basados en Java que permite automatizar la descarga de las librerías necesarias para el funcionamiento de un proyecto.

En este tutorial describiremos cómo instalar Maven, cómo configurarlo para usar el repositorio Artifactory de Sopra Steria, y cómo integrarlo en Eclipse.

## 2. Instalación de Maven

---

Maven se distribuye en forma de binarios precompilados, aunque también está disponible su código fuente. Se puede descargar la última versión desde <https://maven.apache.org/download.cgi>.

**Nota:** las versiones de Maven a partir de la 3.3 necesitan Java JDK 1.7 o superior.

Debemos descargar la distribución binaria que nos interesa y descomprimirla en la carpeta que queramos. Para este tutorial la carpeta de instalación será:

```
D:\Tools\apache-maven-3.5.4
```

Una vez descomprimido debemos configurar las variables de entorno que necesita Maven. Para ello podemos ejecutar las instrucciones siguientes desde una línea de comandos:

```
D:\workspace> setx path %path%;D:\Tools\apache-maven-3.5.4\bin
D:\workspace> setx M2_HOME D:\Tools\apache-maven-3.5.4
```

También es posible configurar estas variables de entorno desde la propiedades del sistema.

Podemos comprobar que Maven está bien instalado ejecutando la instrucción siguiente desde una nueva ventana (las variables de entorno se fijan al abrir la ventana de comandos, los cambios no serán visibles en las que ya estuviesen abiertas, incluyendo la usada para ejecutar las instrucciones anteriores):

```
D:\workspace> mvn -v
```

Esto nos confirmará la versión de Maven instalada y la versión de Java que se está usando.

## 3. Configuración de Maven

---

La configuración de Maven se encuentra en el archivo "settings.xml" que encontraremos en la subcarpeta "conf" de nuestra instalación.

```
D:\Tools\apache-maven-3.5.4\conf\settings.xml
```

Para usar el repositorio central de Maven, únicamente deberemos indicar en este la ruta local en la que descargar las dependencias (el repositorio local), y la configuración del proxy de nuestro site. El contenido por tanto quedará así:

```
<?xml version="1.0" encoding="UTF-8"?>
```



```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">

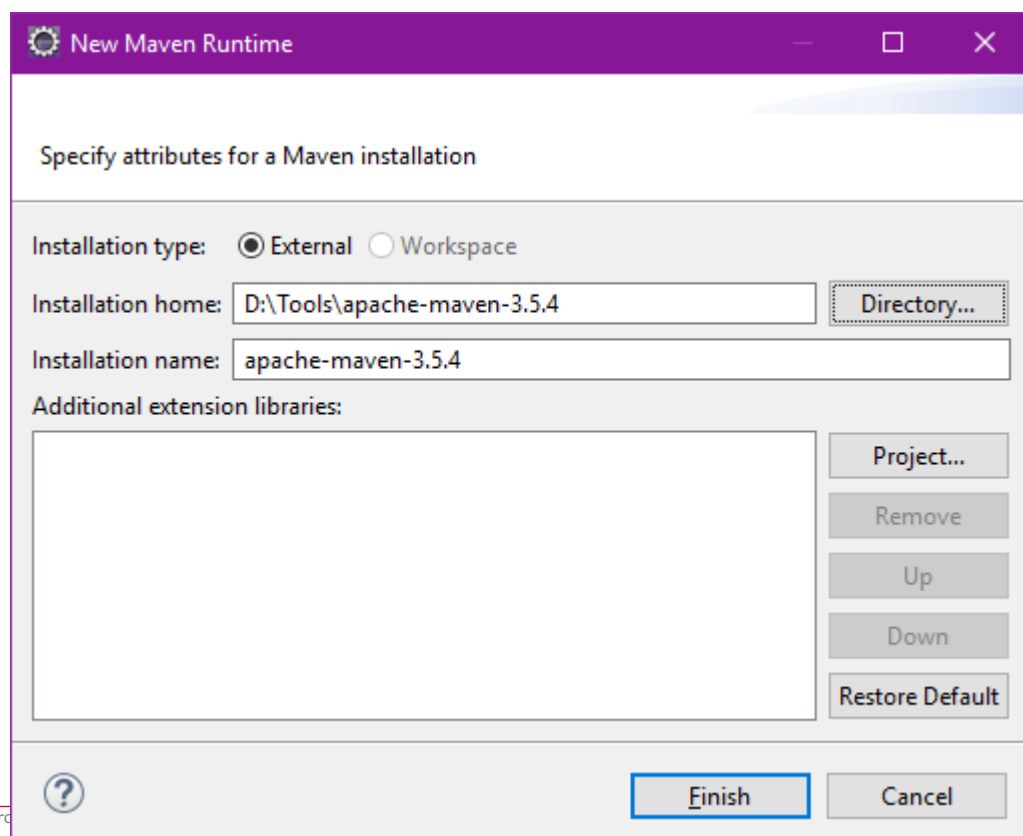
  <localRepository>D:\Tools\Repo</localRepository>

  <proxies>
    <proxy>
      <id>sopra</id>
      <active>true</active>
      <protocol>http</protocol>
      <host>vale.proxy.corp.sopra</host>
      <port>8080</port>
    </proxy>
  </proxies>

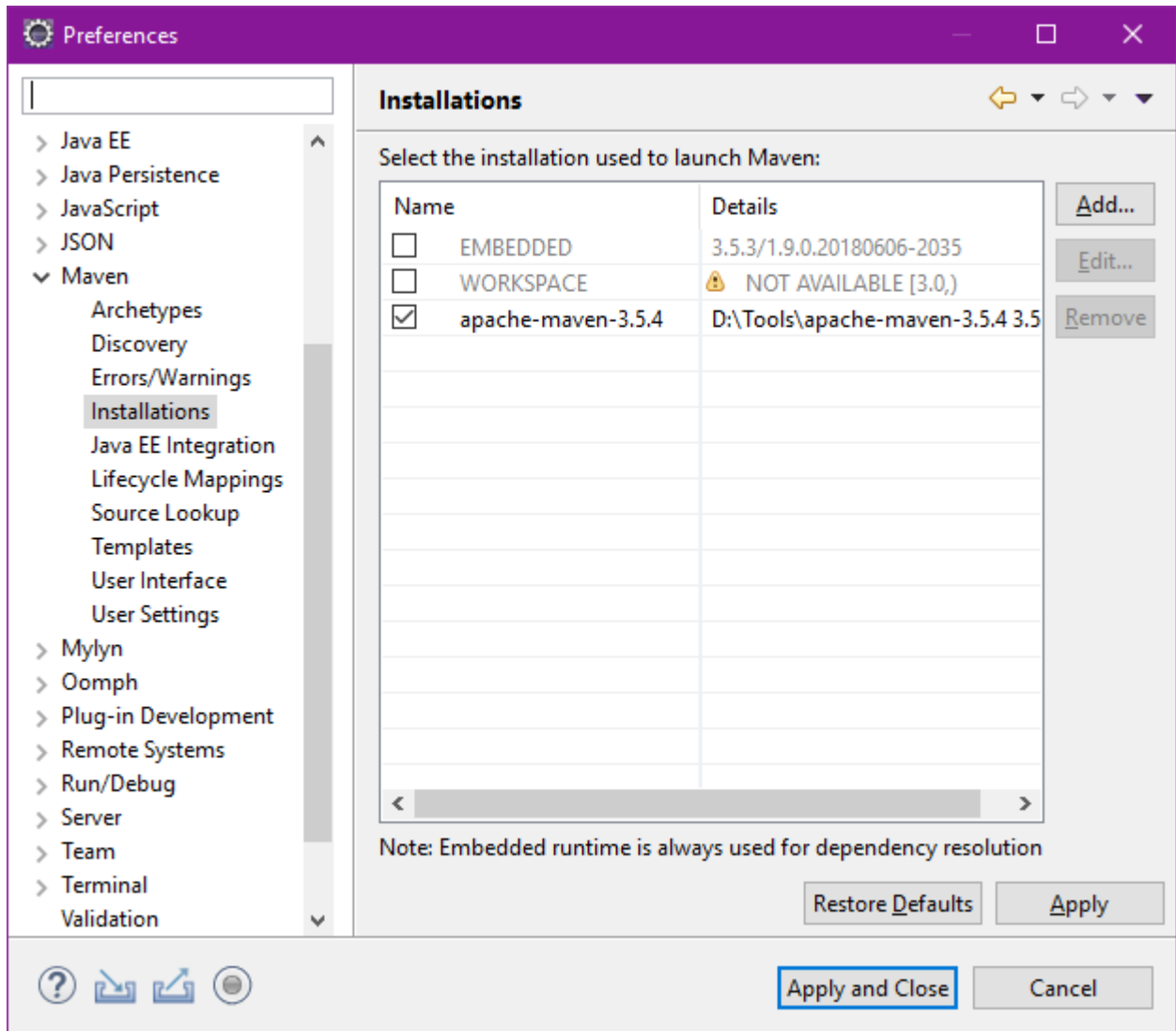
</settings>
```

## 4. Configuración de Maven en Eclipse

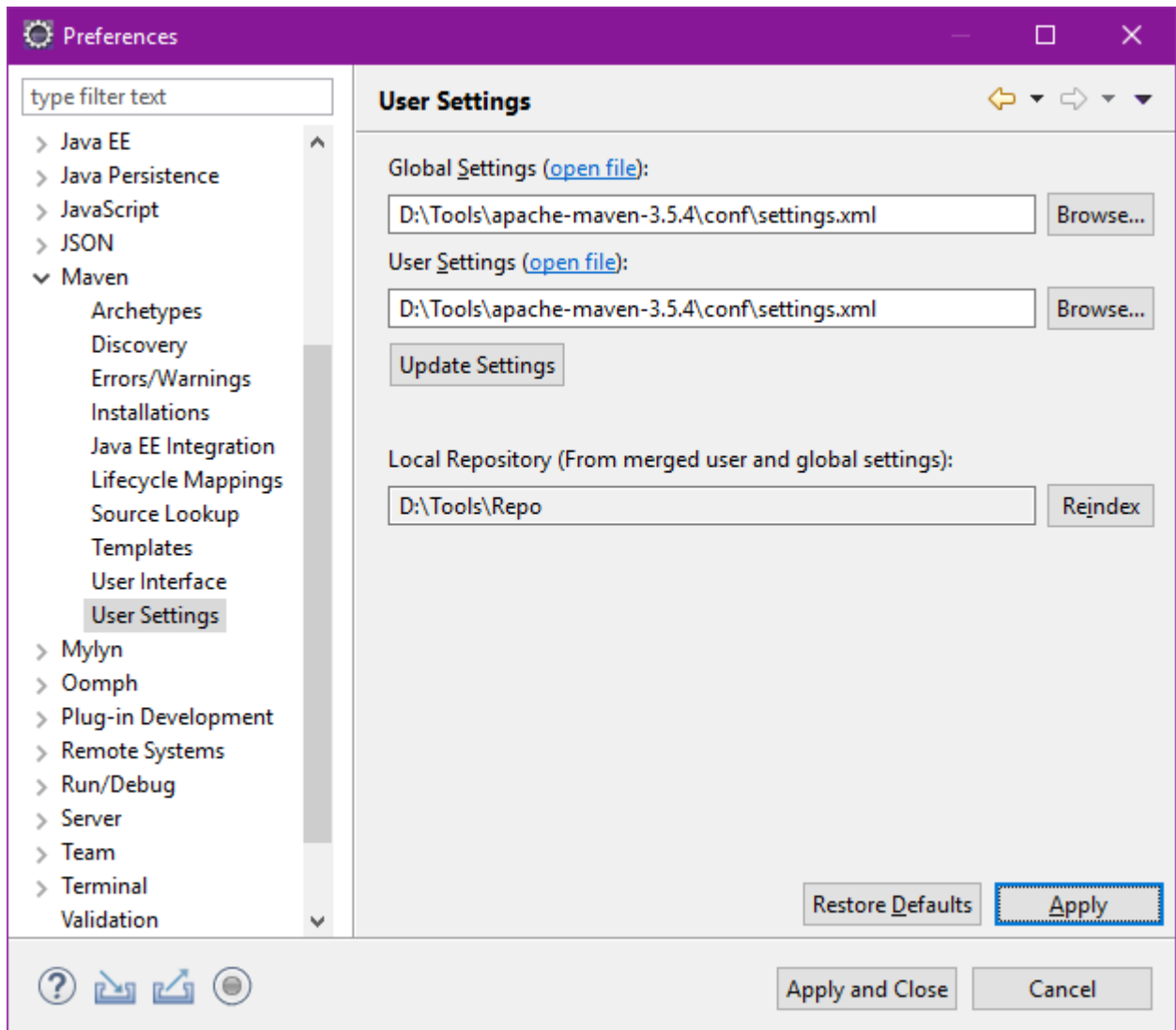
Lanzamos Eclipse y abrimos nuestro workspace. Una vez haya cargado todo, vamos al menú "Window -> Preferences". Ahí navegaremos hasta la categoría "Maven -> Installations". Ahí usaremos la opción "Add..." y la rellenaremos con la ruta de nuestra instalación de Maven:



Una vez añadida marcaremos la casilla a su izquierda para que sea la instalación de Maven que usará Eclipse.



Tras esto deberemos ir a la categoría "Maven -> User Settings". Allí indicaremos el archivo "settings.xml" que hemos construido en el punto anterior en las dos líneas "Global Settings" y "User Settings". Pulsamos sobre "Update Settings" para que Eclipse cargue la nueva configuración, aplicamos cambios, y cerramos.



## 5. Generación de proyectos

Eclipse no consigue resolver correctamente los arquetipos de Maven si intentamos crear un nuevo proyecto directamente desde la interfaz gráfica. Esto podemos solucionarlo invocando a Maven desde línea de comandos. Nos colocaremos en la raíz de nuestro workspace de Eclipse y lanzaremos la orden siguiente:

```
D:\workspace> mvn archetype:generate -DgroupId=com.soprasteria.formacion -
DartifactId=my-app -DarchetypeArtifactId=maven-archetype-webapp -
DinteractiveMode=false
```

Las partes señaladas en amarillo corresponden al identificador de nuestro proyecto, mientras que en verde indicamos el arquetipo a usar. En el ejemplo usamos el que corresponde a aplicaciones web, pero se puede sustituir por otros según necesidad.

Debemos tener cuidado en ejecutar esta orden desde línea de comandos y no desde PowerShell, si nuestro sistema abre PowerShell por defecto (Windows 10), cambiaremos a la línea de comandos clásica mediante la orden `'cmd'`.

Una vez hecho esto podremos importar el proyecto creado en Eclipse seleccionando la opción "Import... -> Maven -> Existing Maven Projects".

