

Tutorial Maven

1.Descargar JDK

link: <https://www.oracle.com/java/technologies/downloads/#java11-windows>
(descargar installer y seguir los pasos)

2.configurar variables de entorno para uso de JDK en eclipse (Conversación con el chatGPT)

A. Configurar `JAVA_HOME` :

1. Encontrar la ruta del JDK:

- Primero, debes saber dónde está instalado el JDK en tu sistema. Por defecto, en Windows, suele estar en una ruta como esta:

```
makefile
```

[Copiar código](#)

```
C:\Program Files\Java\jdk-XX
```

(donde `XX` es la versión del JDK que instalaste, por ejemplo, `jdk-17`).

2. Abrir el panel de Variables de Entorno:

- Haz clic derecho en "Este equipo" (o "Mi PC") y selecciona **Propiedades**.
- En la ventana de **Propiedades del sistema**, haz clic en **Configuración avanzada del sistema** en la barra lateral.
- Luego, haz clic en **Variables de entorno...** en la parte inferior de la nueva ventana.

3. Crear la variable `JAVA_HOME` :

- En la sección **Variables del sistema**, haz clic en **Nueva....**
- En el campo **Nombre de la variable**, escribe `JAVA_HOME` .
- En el campo **Valor de la variable**, pega la ruta de instalación del JDK (por ejemplo, `C:\Program Files\Java\jdk-17`).
- Haz clic en **Aceptar**.

B. Configurar el `PATH` :

1. Editar la variable `Path` :

- En la misma ventana de **Variables de entorno**, busca la variable `Path` en la sección **Variables del sistema**.
- Selecciona `Path` y haz clic en **Editar....**

2. Agregar el JDK al `Path` :

- En la ventana de edición, haz clic en **Nuevo** y agrega la siguiente ruta:

```
perl
```

[Copiar código](#)

```
%JAVA_HOME%\bin
```

- Esto asegura que todos los programas puedan ejecutar los comandos del JDK como `javac` y `java` desde cualquier lugar del sistema.

3. Guardar los cambios:

- Haz clic en **Aceptar** en todas las ventanas abiertas para guardar los cambios.

C. Verificar la configuración:

1. Abrir el Símbolo del Sistema:

- Presiona `Windows + R`, escribe `cmd` y presiona **Enter**.

2. Verificar `JAVA_HOME` :

- Escribe el siguiente comando:

```
bash
```

[Copiar código](#)

```
echo %JAVA_HOME%
```

- Deberías ver la ruta al JDK que configuraste.

3. Verificar el JDK en el `PATH` :

- Escribe:

```
bash
```

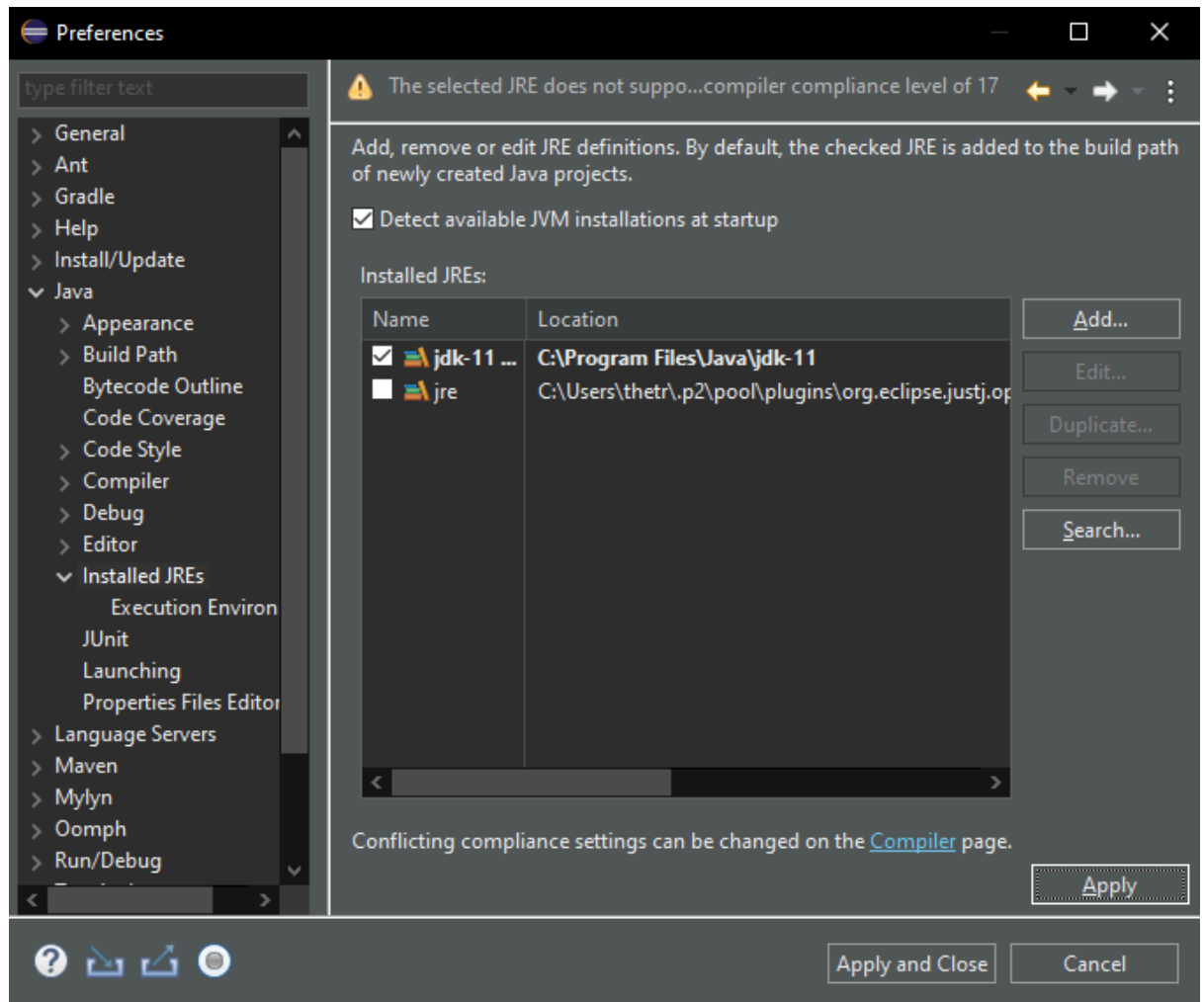
[Copiar código](#)

```
javac -version
```

- Si todo está configurado correctamente, debería mostrarte la versión del compilador de Java (`javac`).

1. Configurar Eclipse para usar el JDK

2. Una vez que tengas el JDK instalado, debes asegurarte de que Eclipse lo use correctamente:
3. Abrir Eclipse.
4. Ve a Window > Preferences (en macOS es Eclipse > Preferences).
5. En el menú de la izquierda, selecciona Java > Installed JREs.
6. Si no ves tu JDK en la lista, haz clic en Add > Standard VM y selecciona la carpeta donde instalaste el JDK.



Una vez tengamos todo esto hecho ya podremos empezar a instalar maven

¿Qué es Maven?

Maven es una herramienta que utilizaremos dentro del propio Eclipse la cual sirve para:

Gestión de dependencias

Estandarización del proceso de construcción: Maven sigue un ciclo de vida de construcción estandarizado que incluye tareas comunes como compilar código, ejecutar pruebas y empaquetar el software.

Ciclo de vida de proyectos: Maneja todo el ciclo de vida de desarrollo de software con fases bien definidas

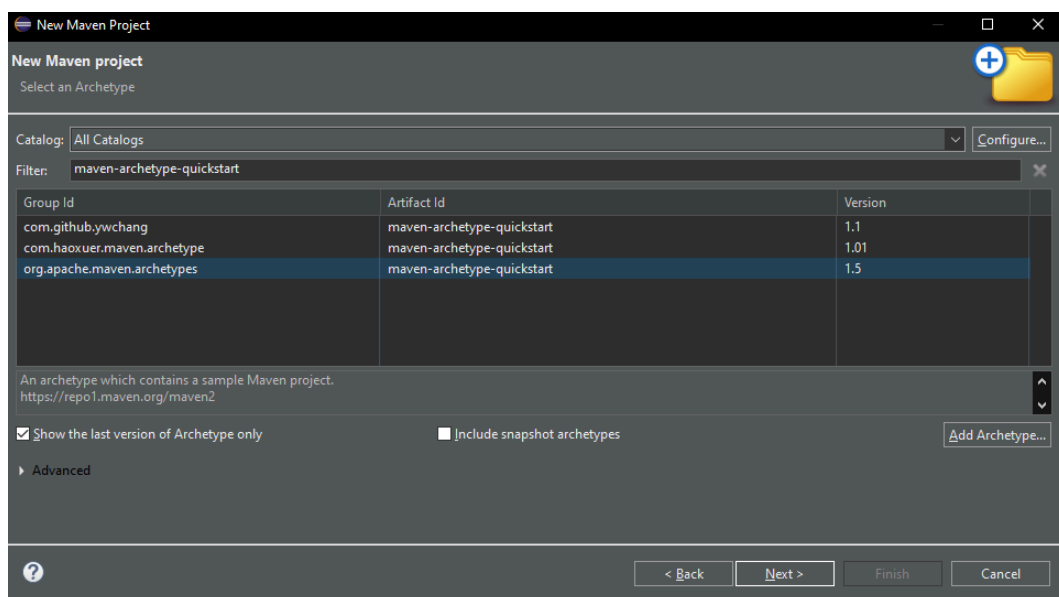
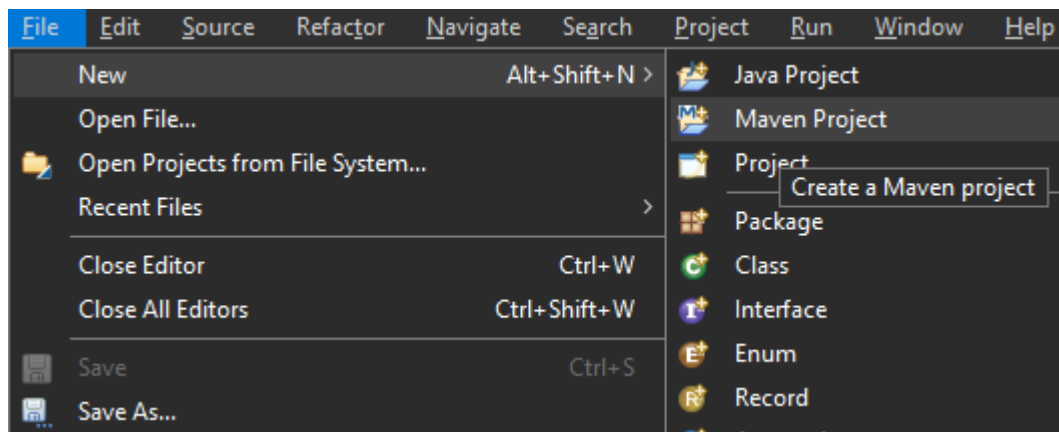
Portabilidad del proyecto: (pom.xml) archivo importante para la portabilidad

Repositorio central y local: Maven utiliza un repositorio central donde almacena las dependencias (bibliotecas y complementos). Además, mantiene un repositorio local en cada máquina donde descarga esas dependencias, para que no sea necesario descargarlas nuevamente en cada construcción.

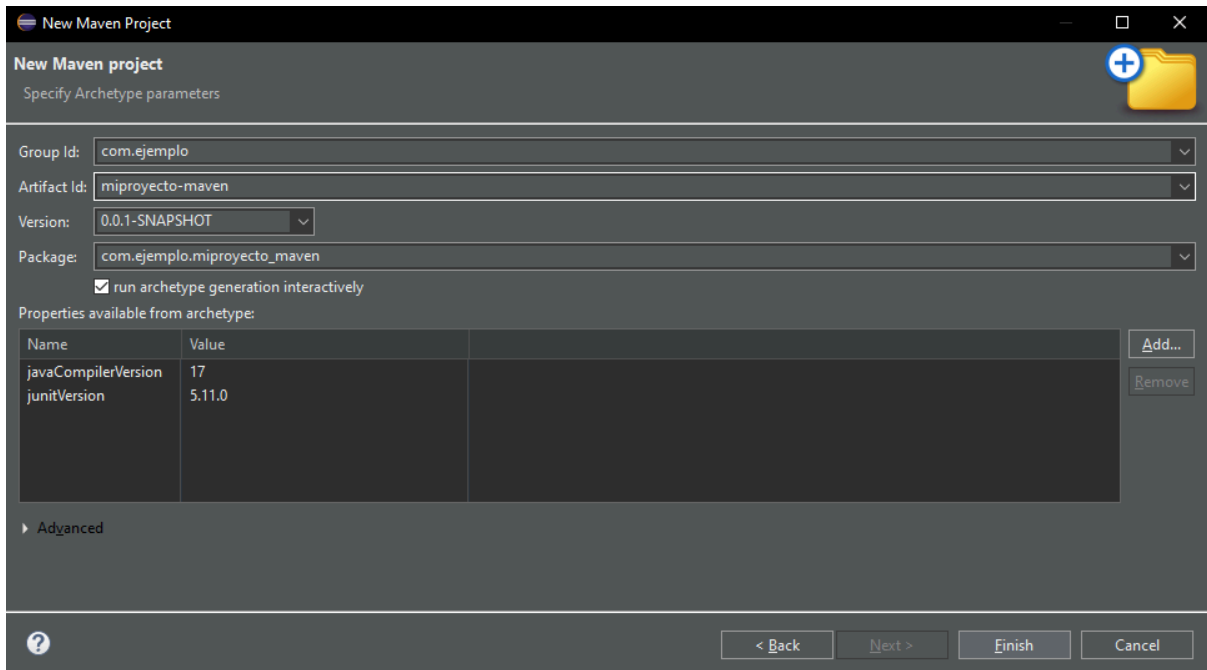
Instalación de Maven

Buenas noticias, si todo va según los expertos en maven no deberías de instalar nada, ya viene nativo en eclipse.

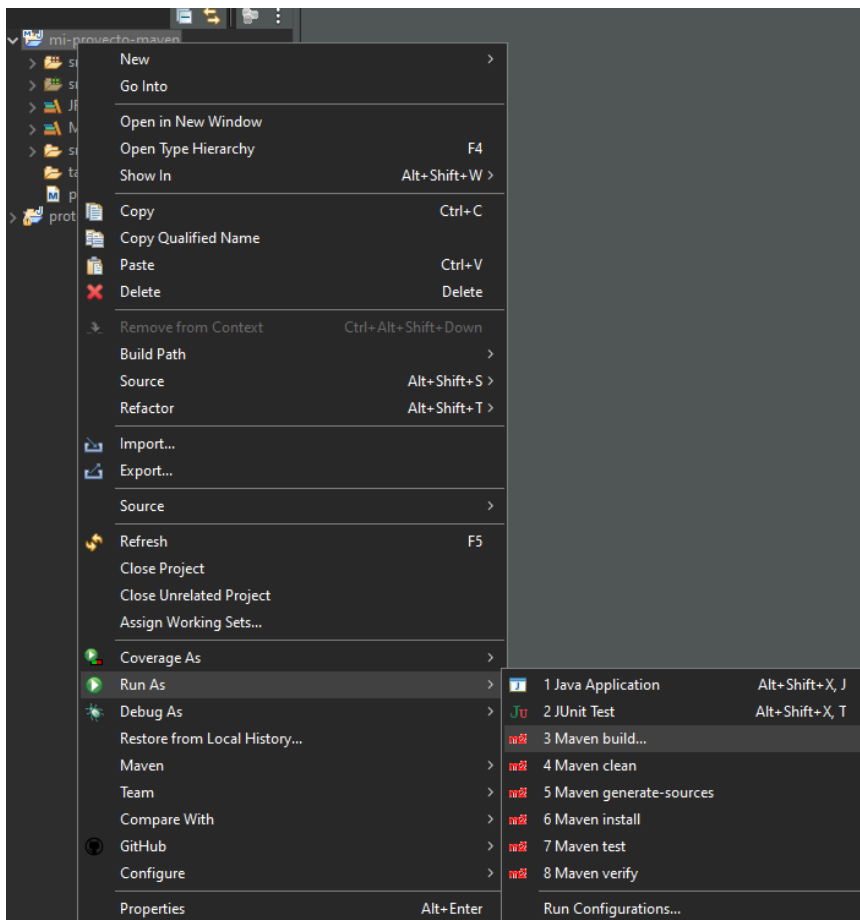
Prueba de proyecto en maven



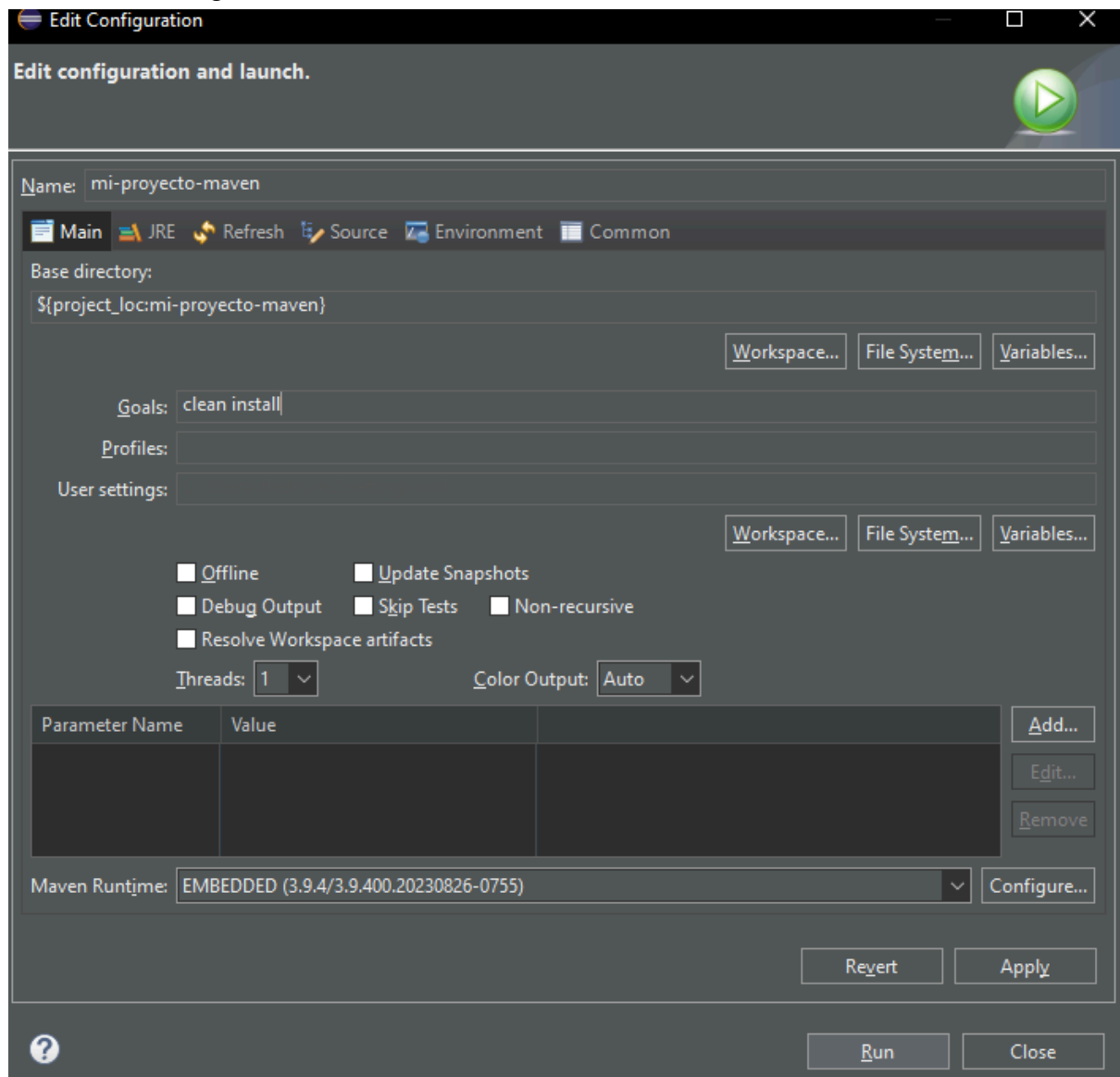
En el filtro poneis esto para la prueba :maven-archetype-quickstart, y seleccionais el oficial de maven.



Cuando le deis a finish, si se os queda “pillado”(se os quedará en un 33%) darle enter en la consola



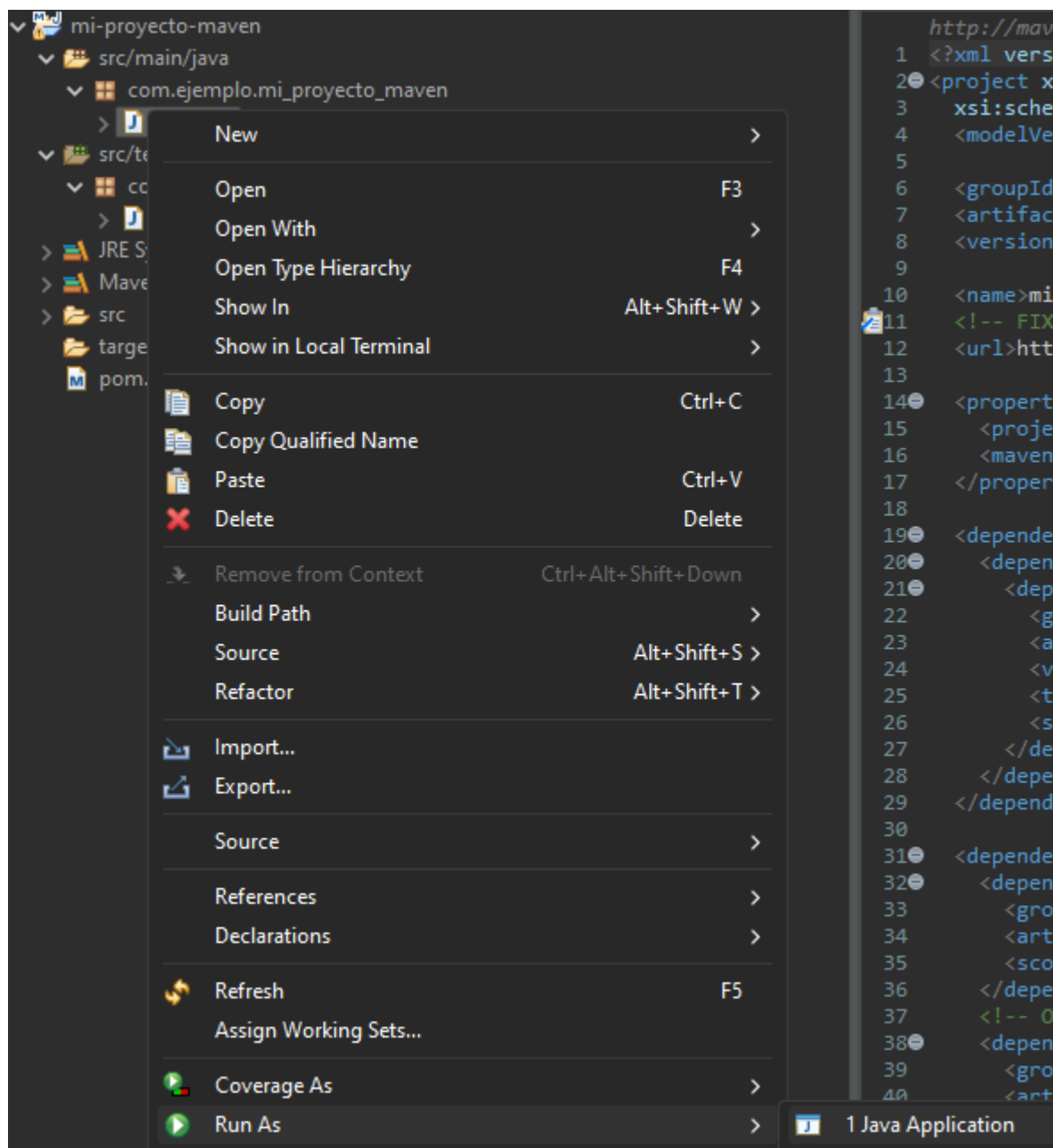
clean install en goals

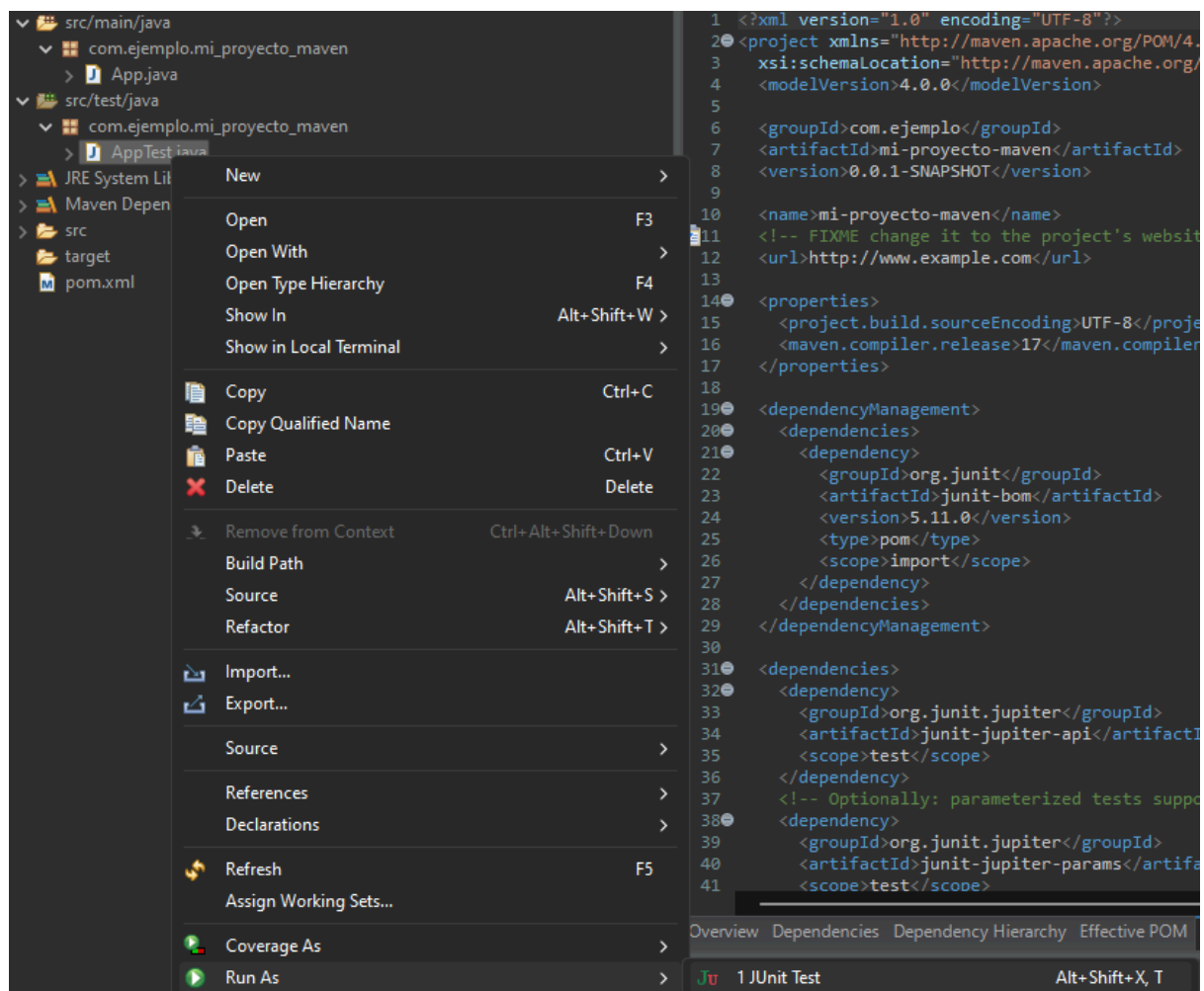


El comando **clean** elimina los archivos compilados antiguos.

El comando **install** compila el proyecto, ejecuta las pruebas y genera los artefactos del proyecto (como el **.jar** o **.war**).

Una vez hecho esto podemos ejecutar estas dos cosas y no debería de darnos ningún error:

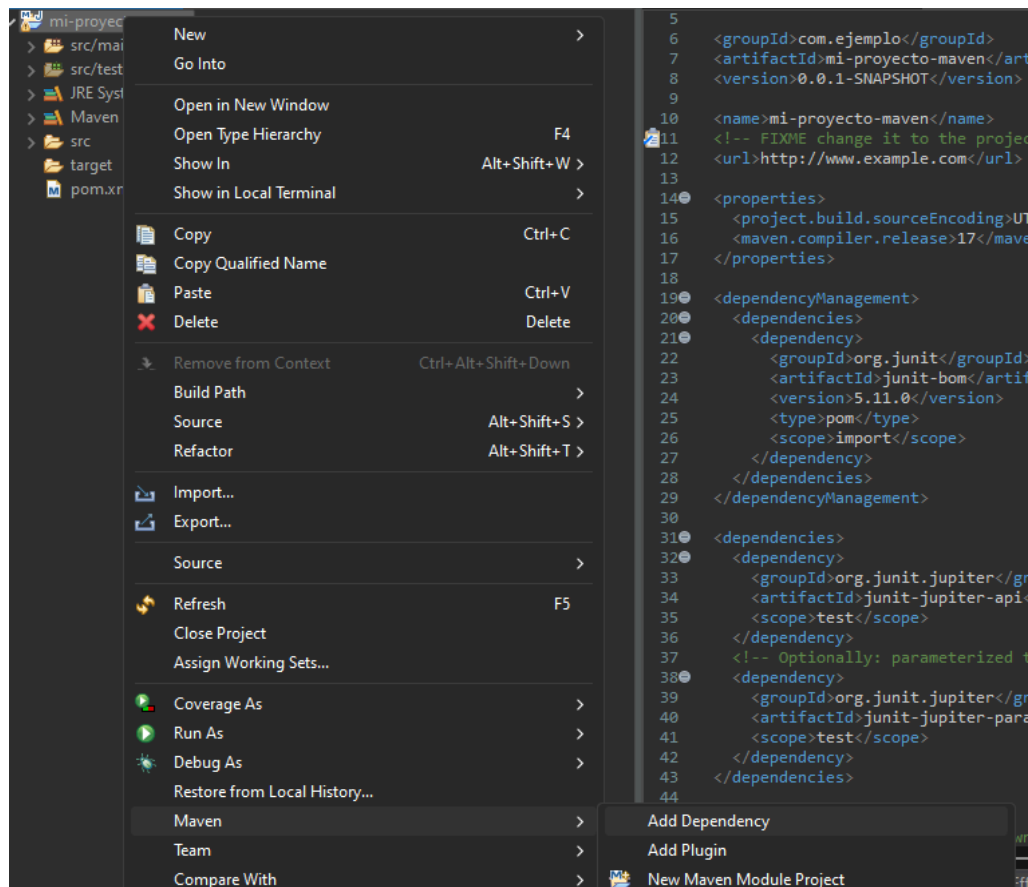




Importancia del POM.xml

Maven utiliza el conocido archivo **POM.xml** (Project Object Model) para dentro de él especificar las diferentes **dependencias o librerías** que serán necesarias incluir en el proyecto que se esté desarrollando.

Para añadir dependencias usando maven se puede hacer añadiendo código al pom.xml o desde la interfaz gráfica de esta forma más sencilla:



Después de agregar las dependencias asegúrate de que el proyecto esté actualizado haciendo clic derecho en el proyecto y seleccionando **Maven > Update Project....**