

Manual Paso a Paso

Gestión y Automatización de Proyectos Java

Francisco José Teurel Teruel

27/01/2026

Índice

Instalación de Maven	1
Creación de un Proyecto	2
3. Estructura del Proyecto	2
4. Compilación	2
5. Gestión de Dependencias	2
6. Ejecución de Pruebas	2
7. Empaquetado	2
8. Instalación en Repositorio Local	2
9. Despliegue en Repositorio Remoto	2
10. Generación de Documentación	2
11. Uso de Perfiles	3
12. Comandos Comunes y Flags	3
Guía Rápida	3
Guía Rápida de Maven (Resumen Práctico)	3
Comandos útiles:	4
Ejemplo Práctico	4
Crear el proyecto:	4
Recursos	5
Conclusión	5

Instalación de Maven

- Instala el JDK y configura JAVA_HOME.
- Descarga y descomprime Maven.
- Añade el directorio bin de Maven al PATH.
- Verifica la instalación con mvn -version.

Creación de un Proyecto

```
mvn archetype:generate -DgroupId=com.ejemplo -DartifactId=demo-maven  
-DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

3. Estructura del Proyecto

- `src/main/java`: Código fuente.
- `src/main/resources`: Recursos.
- `src/test/java`: Pruebas.
- `pom.xml`: Configuración central.

4. Compilación

```
mvn compile
```

5. Gestión de Dependencias

- Añade dependencias en `<dependencies>` del POM.
- Ejecuta `mvn dependency:tree` para visualizar el árbol.

6. Ejecución de Pruebas

```
mvn test
```

7. Empaquetado

```
mvn package
```

8. Instalación en Repositorio Local

```
mvn install
```

9. Despliegue en Repositorio Remoto

- Configura `<distributionManagement>` y credenciales en `settings.xml`.
- Ejecuta `mvn deploy`.

10. Generación de Documentación

```
mvn site
```

11. Uso de Perfiles

- Define perfiles en <profiles> del POM.
- Activa con mvn install -P<perfil>.

12. Comandos Comunes y Flags

Comando	Descripción
mvn clean	Limpia el directorio target.
mvn compile	Compila el código fuente.
mvn test	Ejecuta pruebas unitarias.
mvn package	Empaqueña el proyecto.
mvn install	Instala en repositorio local.
mvn deploy	Despliega en repositorio remoto.
mvn site	Genera documentación del sitio.
mvn dependency:tree	Muestra árbol de dependencias.
mvn help:effective-pom	Muestra POM efectivo.
mvn -X	Modo debug detallado.
mvn -T 4	Usa 4 hilos para build.
mvn -o	Modo offline.
mvn -DskipTests	Omite ejecución de pruebas.

Guía Rápida

Guía Rápida de Maven (Resumen Práctico)

Instala Java (JDK) y Maven.

Crea un proyecto básico:

```
mvn archetype:generate -DgroupId=com.ejemplo -DartifactId=mi-app  
-DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

Compila el proyecto:

```
mvn compile
```

Ejecuta pruebas:

```
mvn test
```

Empaqueña el proyecto:

```
mvn package
```

Instala en el repositorio local:

```
mvn install
```

Agrega dependencias: Edita pom.xml en la sección <dependencies>.

Comandos útiles:

- Limpiar: mvn clean
- Árbol de dependencias: mvn dependency:tree
- Modo offline: mvn -o
- Omite pruebas: mvn -DskipTests package
- Importa el proyecto en tu IDE (IntelliJ, Eclipse, VS Code).
- Consulta la documentación oficial: <https://maven.apache.org/>

Ejemplo Práctico

Crear el proyecto:

```
mvn archetype:generate -DgroupId=com.ejemplo -DartifactId=demo-maven  
-DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

Estructura generada:

```
demo-maven/  
|---- pom.xml  
|---- src/  
|----- main/java/com/ejemplo/App.java  
|----- test/java/com/ejemplo/AppTest.java
```

Compilar y probar:

```
cd demo-maven  
mvn clean compile  
mvn test
```

Añadir una dependencia (ejemplo: Gson): En pom.xml:

```
<dependency>  
  <groupId>com.google.code.gson</groupId>  
  <artifactId>gson</artifactId>  
  <version>2.10.1</version>  
</dependency>
```

Empaquetar:

```
mvn package
```

Ejecutar la aplicación (si tiene método main):

```
mvn exec:java -Dexec.mainClass="com.ejemplo.App"
```

Recursos

- Documentación oficial de Maven
- Maven Central Repository
- Guía de instalación de Maven
- Guía de comandos Maven
- Comparativa Maven vs Gradle
- Solución de problemas comunes
- Generación de sitios y documentación

Conclusión

Apache Maven es una herramienta esencial para el desarrollo profesional en Java, proporcionando automatización, estandarización y gestión eficiente de dependencias y builds. Su curva de aprendizaje se ve compensada por la robustez, la integración con el ecosistema Java y la facilidad para mantener proyectos a largo plazo. Dominar Maven es una inversión clave para cualquier desarrollador o equipo que busque calidad, reproducibilidad y escalabilidad en sus proyectos de software.