PROMETEO

PROGRAMACIÓN

MAVEN

¿Qué es Maven?

Definición:

- Herramienta de gestión y automatización de proyectos
- Basada en XML
- Especialmente usada en proyectos Java

Objetivo principal:

- Gestionar dependencias
- Estandarizar estructura
- Automatizar compilación, test y empaquetado

¿Por qué surge Maven?

Problemas antes de Maven:

- Gestión manual de librerías (.jar)
- Builds diferentes en cada equipo
- Scripts Ant complejos

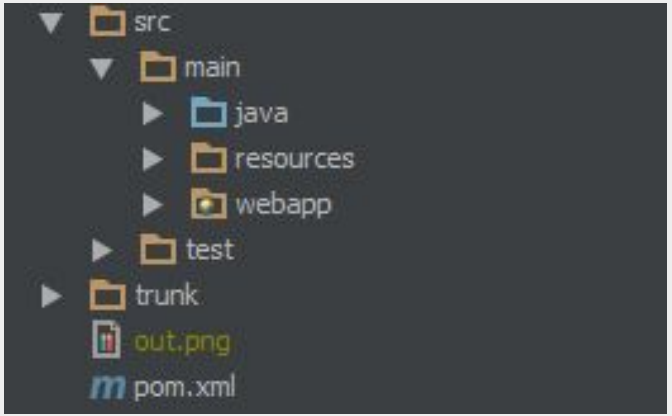
Solución que aporta Maven:

- Convención sobre configuración
- Repositorio centralizado
- Ciclo de vida estandarizado

Estructura

La estructura estándar de un proyecto Maven

```
mi-proyecto/
├── src/
│   ├── main/
│   │   ├── java/
│   │   └── resources/
│   └── test/
│       └── java/
└── pom.xml
```



El archivo POM.xml

El **POM (Project Object Model)** es el archivo principal de configuración de un proyecto Maven.

Es:

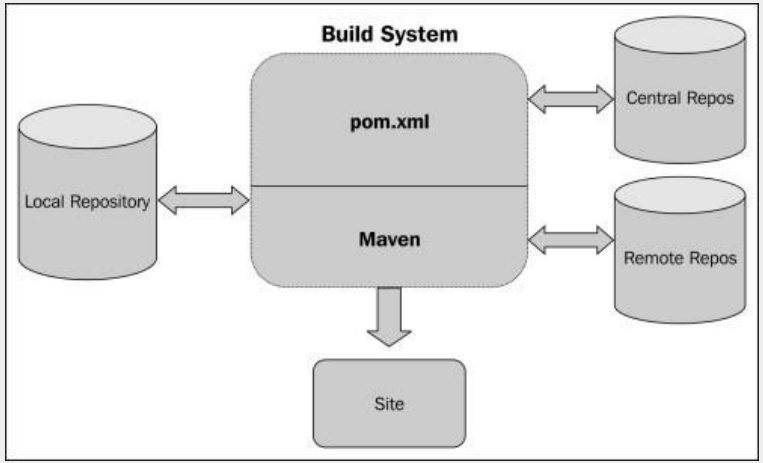
- Obligatorio en todo proyecto Maven
- El “cerebro” del proyecto

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0">

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.ejemplo</groupId>
    <artifactId>mi-proyecto</artifactId>
    <version>1.0.0</version>

</project>
```



Dependencias

Uno de los elementos más importantes del POM:

```xml
<dependencies>
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter</artifactId>
        <version>5.9.0</version>
        <scope>test</scope>
    </dependency>
</dependencies>
```

¿Qué hace esto?

- ✔ Descarga automáticamente la librería
- ✔ La guarda en el repositorio local
- ✔ La añade al classpath del proyecto

Sección <build>

Permite configurar cómo se construye el proyecto.

```
<build>
    <plugins>
        <plugin>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.11.0</version>
            <configuration>
                <source>17</source>
                <target>17</target>
            </configuration>
        </plugin>
    </plugins>
</build>
```

Aquí estamos indicando:

- Versión de Java
- Cómo compilar el proyecto

Herencia en Maven

El POM puede heredar de otro POM

```xml
<parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.2.0</version>
</parent>
```

Esto permite:

- No repetir configuraciones
- Gestionar versiones centralizadas
- Simplificar el proyecto

Muy usado en proyectos Spring.

Otras secciones importantes

properties

Permite definir variables reutilizables

```
<properties>
    <java.version>17</java.version>
</properties>
```

packaging

Da formato a la compilación

```
<packaging>jar</packaging>
```

Ventajas y Desventajas

✅ Ventajas

- Automatización
- Gestión centralizada
- Estándar en proyectos Java
- Integración con IDEs (IntelliJ, Eclipse, VS Code)

❌ Desventajas

- XML verboso
- Curva de aprendizaje
- Puede ser rígido

Conclusión

- ★ Maven estandariza proyectos Java
- ★ Automatiza procesos repetitivos
- ★ Facilita el trabajo en equipo
- ★ Es una herramienta fundamental en entornos profesionales


PROMETEO