



Ing. Max Cerna

---

# Maven

Herramienta de gestión y comprensión de proyectos que se utiliza principalmente en proyectos Java.

# Principales Características

## 1. Gestión de Dependencias

Simplifica la gestión de las bibliotecas y frameworks que un proyecto necesita.

archivo de configuración llamado ***pom.xml*** (Project Object Model)

descarga automáticamente estas dependencias

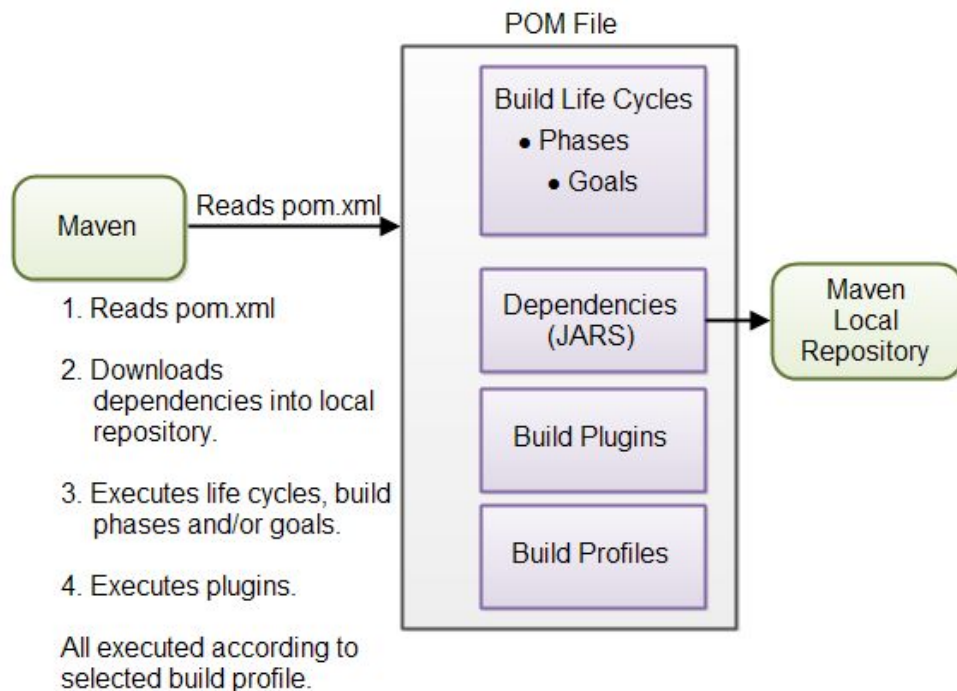
# Principales Características

## 2. Construcción del Proyecto

Proporciona un marco de construcción que puede:

- compilar
- ejecutar pruebas
- empaquetar el código en formatos como JAR o WAR

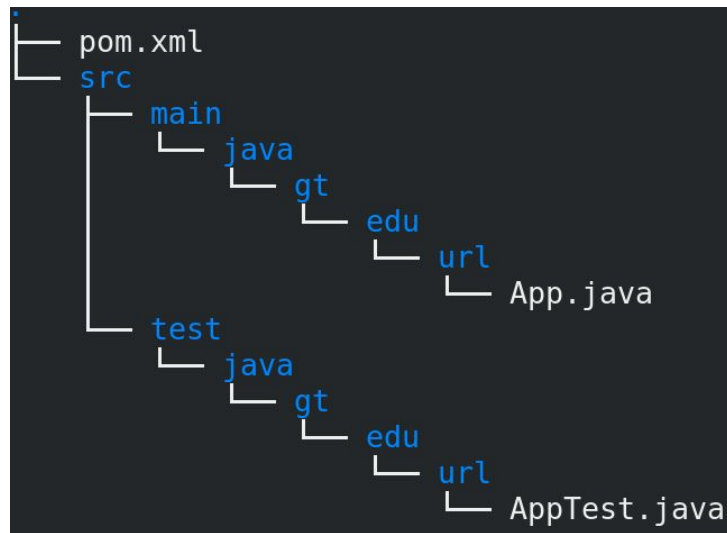
# Principales Características



# Principales Características

## 3. Estructura del Proyecto

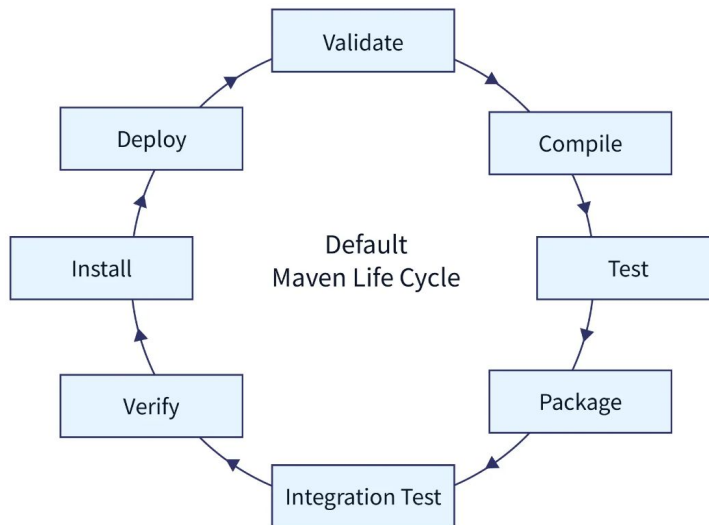
Promueve una estructura de proyecto estándar que facilita la comprensión y organización del código.



# Principales Características

## 4. Ciclo de Vida de Construcción

Define un ciclo de vida de construcción que consta de una serie de fases.



# Principales Características

- **clean** - Elimina el directorio target
- **validate** - Valida si el proyecto es correcto
- **compile** - Compila los archivos fuente y almacena el resultado en **target/classes**
- **test** - Ejecuta tests
- **package** - Toma los objetos compilados y los empaca en formatos distribuibles e.g. JAR, WAR
- **verify** - Ejecuta tests adicionales sobre el paquete para control de calidad
- **install** - Instala el paquete en el repositorio local
- **deploy** - Despliega el paquete en un repositorio remoto



# Principales Características

## 5. **Plugins y Extensibilidad**

Utiliza un sistema de plugins que permite extender su funcionalidad.

## 6. **Repositorios**

Trabaja con repositorios locales y remotos para gestionar dependencias y artefactos del proyecto.

# Maven - Arquetipo

Conjunto de plantillas que proporcionan una estructura de proyecto predefinida.

Esto incluye directorios estándar, archivos de configuración (pom.xml), código fuente de ejemplo, archivos de prueba, y cualquier otro recurso necesario.

Algunos ejemplos comunes incluyen:

- **maven-archetype-quickstart**: Crea un proyecto Java básico con una estructura mínima.
- **maven-archetype-webapp**: Crea un proyecto para una aplicación web basada en Java.

# Maven - Arquetipo

```
mvn archetype:generate -DgroupId=com.example -DartifactId=my-app \
-DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

- DgroupId: Especifica el identificador del grupo para el proyecto hacia una empresa/organización.
- DartifactId: Especifica el identificador del artefacto para el proyecto.
- DarchetypeArtifactId: Especifica el arquetipo que se utilizará (en este caso, maven-archetype-quickstart).
- DinteractiveMode=false: Desactiva el modo interactivo, permitiendo que el comando se ejecute sin pedir confirmación para cada paso.

# Maven - Arquetipo

## Sin arquetipo

```
mvn archetype:generate -DgroupId=gt.edu.url -DartifactId=demo1
```

## con arquetipo

```
mvn archetype:generate -DgroupId=gt.edu.url -DarchetypeArtifactId=maven-archetype -quickstart -DartifactId=demo1
```

# Instalacion Windows

- 1) Ir a <https://maven.apache.org/download.cgi>
- 2) Descargar el archivo binario ZIP de la última versión de Maven.
- 3) Extraer el contenido del archivo ZIP a un directorio de tu elección,

por ejemplo, **C:\Program Files\Apache\Maven**

- 4) Abrir el Panel de Control y navegar a **Sistema > Configuración avanzada del sistema > Variables de entorno.**

Nombre de la variable: **MAVEN\_HOME**

Valor de la variable: **C:\Program Files\Apache\Maven**

Encuentrar la variable Path en las Variables del sistema. Agrega %MAVEN\_HOME%\bin al final de la lista de rutas.

- 5) Abrir una ventana de comandos (CMD) y escribir **mvn -version**