

Fases Principales de Maven

Ciclos de vida y su relación entre ellas

Presentado por Equipo de Desarrollo

Resumen

Maven es una herramienta de automatización de construcción basada en el concepto de ciclos de vida. Esta presentación explora los tres ciclos de vida fundamentales (default, clean y site), detallando sus fases principales y la relación jerárquica entre fases y goals. Comprenderemos cómo estos componentes trabajan juntos para estandarizar y automatizar el proceso de construcción de software en proyectos Java.

↳ CONCEPTOS FUNDAMENTALES

Un ciclo de vida en Maven está compuesto por fases secuenciales, y cada fase agrupa uno o más goals. Entender esta relación es esencial para configurar correctamente el archivo POM y ejecutar comandos Maven eficientemente, permitiendo la automatización completa del proceso de construcción.

Índice

Exploración completa sobre los ciclos de vida de Maven, sus fases principales y la relación entre ellas para comprender el flujo de construcción de proyectos Java.

Introducción a
Maven

Ciclos de vida

Ciclo Default

Ciclo Clean

Ciclo Site

Fases principales

Goals y Plugins

Relación fases-
goals

Comandos comunes

Ejecución de
fases

Personalización

Conclusiones

¿Qué es Maven?

Maven es una herramienta de automatización y gestión de proyectos Java que estandariza el proceso de construcción mediante ciclos de vida bien definidos.

Herramienta de Automatización

Maven simplifica y estandariza el proceso de construcción de software, permitiendo a los desarrolladores enfocarse en la lógica del negocio en lugar de los detalles de construcción. Proporciona una estructura de proyecto uniforme y gestión centralizada de dependencias.

Importancia de los Ciclos de Vida

Los ciclos de vida definen secuencias específicas de fases de construcción que se ejecutan en orden predefinido. Esta estructura garantiza consistencia, facilita la automatización de procesos y permite que cualquier desarrollador pueda entender y ejecutar el proyecto con comandos simples.

Ciclos de Vida

Maven define tres ciclos de vida principales, cada uno con sus propias fases y propósitos específicos. Estos ciclos de vida están diseñados para manejar diferentes aspectos del proceso de construcción de software, desde la compilación y empaquetado hasta la limpieza y documentación.

Default Lifecycle

Es el ciclo principal responsable de la construcción y despliegue del proyecto. Incluye las fases más importantes como `validate`, `compile`, `test`, `package`, `verify`, `install` y `deploy`. Este ciclo maneja todo el proceso desde la compilación hasta la distribución.

Clean Lifecycle

Se encarga de limpiar los archivos generados por construcciones anteriores. Consta de solo tres fases: `pre-clean`, `clean` y `post-clean`. Se ejecuta de forma independiente para preparar un entorno limpio antes de una nueva compilación.

Site Lifecycle

Dedicado a la creación y despliegue de la documentación del proyecto. Incluye las fases `pre-site`, `site`, `post-site` y `site-deploy`. Es fundamental para mantener la documentación actualizada junto con el código del proyecto.

Estructura

Cada ciclo de vida en Maven está compuesto por fases secuenciales que se ejecutan en un orden predefinido, permitiendo automatizar completamente el proceso de construcción y gestión del proyecto.

Fases Secuenciales

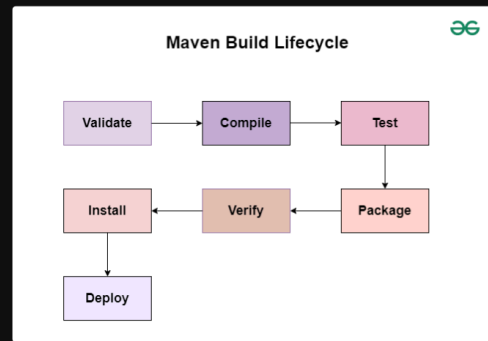
Las fases son etapas discretas en el ciclo de vida de la construcción, ejecutadas en un orden específico. Al invocar una fase concreta (ej. `mvn package`), Maven ejecuta automáticamente todas las fases anteriores hasta llegar a la fase solicitada, garantizando una construcción completa del proyecto.

Composición de los Ciclos

Los tres ciclos de vida (`default`, `clean`, `site`) tienen diferentes fases según su propósito. El ciclo `default` posee 23 fases, `clean` solamente 3 fases, y `site` tiene 4 fases. Esta estructura modular permite combinar comandos como `"mvn clean install"` para ejecutar secuencias completas de operaciones.

Fases Principales del Ciclo Default

El ciclo de vida default es el principal en Maven y se encarga de todo el proceso de construcción, desde la validación hasta el despliegue. Cada fase representa una etapa específica que debe completarse secuencialmente.



1 Validate → Compile → Test

Validate: Verifica que el proyecto es correcto y toda la información necesaria está disponible.

Compile: Compila el código fuente del proyecto.

Test: Ejecuta las pruebas unitarias utilizando un framework adecuado.

2 Package → Verify → Install → Deploy

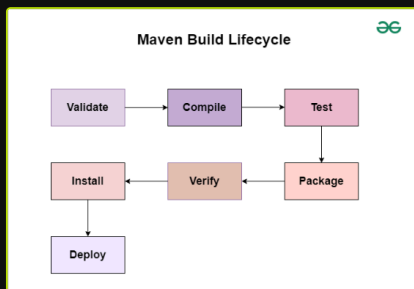
Package: Empaqueta el código compilado en formato distribuible (JAR, WAR, etc.).

Verify: Ejecuta verificaciones para asegurar que el paquete cumple criterios de calidad.

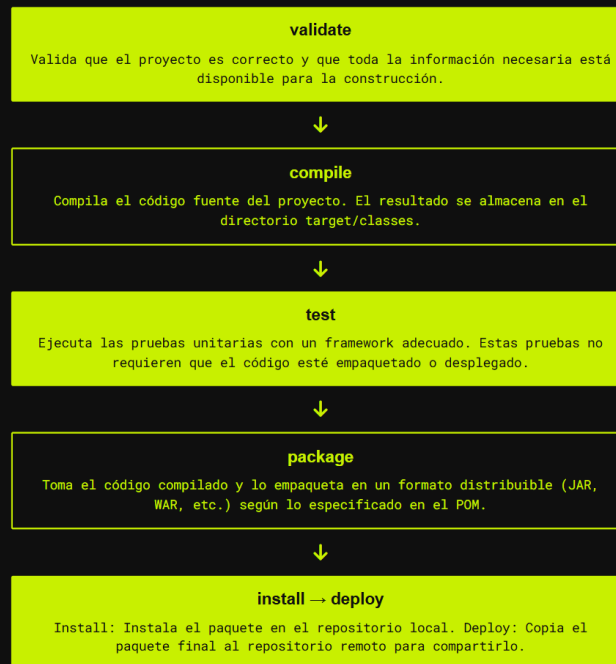
Install: Instala el paquete en el repositorio local para uso como dependencia.

Deploy: Copia el paquete final al repositorio remoto para compartirlo.

Visualización del flujo de fases



Las fases se ejecutan secuencialmente. Al ejecutar una fase, todas las anteriores se ejecutan en orden.



Ciclos Clean & Site

Además del ciclo default, Maven ofrece dos ciclos de vida auxiliares pero fundamentales: `clean` para eliminar archivos generados en compilaciones previas y `site` para crear la documentación del proyecto. Cada uno tiene sus propias fases y propósitos específicos.

Ciclo Clean

Encargado de eliminar todos los archivos generados en compilaciones previas, como clases compiladas, recursos procesados y archivos JAR/WAR. Permite comenzar desde un estado limpio.

`pre-clean` `clean` `post-clean`

Ciclo Site

Se utiliza para generar la documentación del proyecto, informes y sitios web. Fundamental para la documentación técnica y la comunicación del proyecto.

`pre-site` `site` `post-site` `site-deploy`

Uso Combinado

Los ciclos se pueden ejecutar juntos: `mvn clean site` primero elimina los archivos generados anteriormente y luego genera la documentación actualizada, asegurando información precisa y completa.

Fases y Goals

En Maven, cada fase está vinculada a uno o más goals, que son tareas específicas ejecutadas por los plugins. Esta vinculación (binding) es fundamental para entender el funcionamiento interno del ciclo de vida de Maven.

¿Qué son los Goals?

Los goals son unidades individuales de trabajo que realizan los plugins de Maven. Por ejemplo, `compiler:compile` es un goal del plugin `compiler`, que compila el código fuente del proyecto. Los goals son más granulares que las fases y se encargan de tareas específicas como compilar, ejecutar tests, generar documentación, etc.

Binding de Goals a Fases

El binding puede ser:

- **Por defecto:** Según el tipo de packaging (`jar`, `war`, etc.), Maven asigna goals específicos a fases.
- **Explícito:** Configurado en el POM, permitiendo personalizar qué goals se ejecutan en cada fase.

Múltiples goals pueden estar vinculados a una misma fase, ejecutándose en el orden definido.

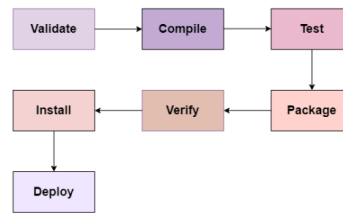
Ejemplo práctico

El comando `mvn clean install deploy` ejecuta una secuencia completa de fases encadenando múltiples ciclos de vida. Representa un flujo de trabajo común para limpiar, construir y publicar un proyecto Java.

```
$ mvn clean install deploy
```

Este comando combinado es extremadamente útil para procesos de integración continua y despliegue, ya que ejecuta una secuencia completa y ordenada de operaciones con un solo comando.

Maven Build Lifecycle



1 Ciclo clean

Primero se ejecuta el ciclo completo de `clean`, que incluye las fases `pre-clean`, `clean` y `post-clean`. Esto elimina todos los archivos generados en compilaciones anteriores, como las carpetas `target/` y los archivos `.class`, preparando un entorno limpio.

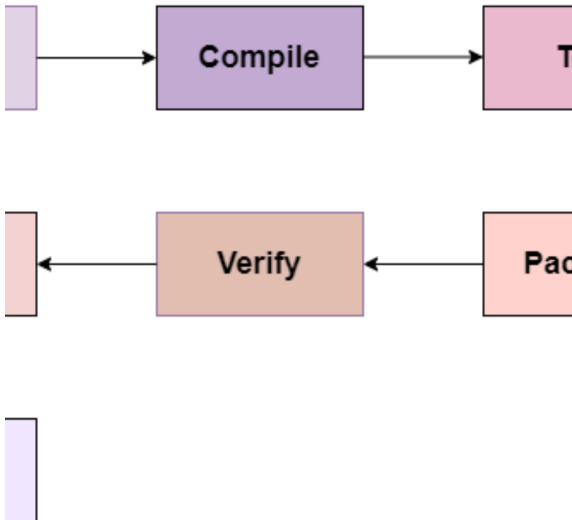
2 Ciclo default (hasta install)

A continuación, se ejecuta el ciclo default hasta la fase `install`. Esto incluye `validate`, `compile`, `test`, `package` e `install`, que compila el código, ejecuta las pruebas, empaqueta y guarda el artefacto en el repositorio local de Maven.

3 Fase deploy

Finalmente, se ejecuta la fase `deploy`, que toma el artefacto generado y lo publica en un repositorio remoto (como Nexus o Artifactory) para que esté disponible para otros desarrolladores o sistemas.

Maven Build Lifecycle



Conclu siones

Entender las fases de Maven y su relación permite optimizar los procesos de construcción y despliegue de proyectos Java. El conocimiento de los ciclos de vida proporciona mayor control sobre las etapas de desarrollo, aumentando la productividad y estandarizando los procesos de integración continua. La correcta aplicación de comandos como `'mvn clean install'` o `'mvn verify'` asegura la calidad del código y reduce errores en producción.