

■ Maven Internals — Módulo 1: Arquitectura Interna

Comprender los componentes internos de Maven —cómo se comunican y qué responsabilidades tiene cada uno— para entender qué pasa cuando ejecutas mvn clean install.

1■■■ Project Object Model (POM) & maven-model-builder

• Combina pom.xml, , BOMs, perfiles, propiedades.
• Produce un Model interpolado con herencia y overrides.
• Usa maven-model-builder para fusionar padres y perfiles.
• Salida: instancia de Model usada por el resto del pipeline.

2■■■ maven-core, MavenSession & LifecycleExecutor

• maven-core inicia MavenSession, Settings, Toolchains.
• LifecycleExecutor mapea fases a goals de plugins (compile, test, package...).
• Ejecuta los ciclos: validate → compile → test → package → verify → install → deploy.

3■■■ Reactor (multi-módulo)

• Lee del POM raíz y construye un grafo de dependencias internas.
• Ordena en topología y soporta ejecución paralela (-T 1C).
• Cada módulo se construye con su propio MavenProject.

4■■■ Artifact Resolver (Aether)

• Resuelve dependencias desde ~/.m2/repository y repos remotos (Central, Nexus, Artifactory).
• Verifica checksums, mantiene caché local y aplica políticas de actualización.
• Conflictos: nearest-wins, exclusiones, dependencyManagement para forzar versiones.

5■■■ Plugin Manager & Mojos

• Localiza plugins y lee plugin.xml.
• Inyecta contexto (Session, Project, Settings).
• Ejecuta Mojo.execute() para cada goal (compile, test, jar...).

6■■ Artefactos & Repositorios

- | |
|---|
| • Salida: JAR/WAR + POM efectivo + metadatos (checksums, maven-metadata.xml). |
| • install → repo local; deploy → repo remoto (Nexus/Artifactory). |

Notas clave:

- Herencia & perfiles: maven-model-builder compone el modelo efectivo antes de ejecutar.
- Reactor: calcula orden topológico entre módulos y permite paralelismo.
- Resolver: usa Aether y nearest-wins en conflictos de versión.