

Ampliación de Ingeniería del Software

Anexo – Uso de Maven por terminal



Universidad
Rey Juan Carlos

Micael Gallego

Correo: micael.gallego@urjc.es
Twitter: [@micael_gallego](https://twitter.com/micael_gallego)

Francisco Gortázar

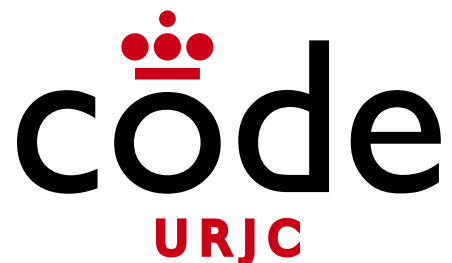
Correo: francisco.gortazar@urjc.es
Twitter: [@fgortazar](https://twitter.com/fgortazar)

Michel Maes

michel.maes@urjc.es

Óscar Soto

oscar.soto@urjc.es



©2023

Micael Gallego, Francisco Gortázar, Michel Maes, Óscar Soto

Algunos derechos reservados

Este documento se distribuye bajo la licencia
“Atribución-CompartirIgual 4.0 Internacional”
de Creative Commons Disponible en
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

Maven

¿Qué es Maven?

- Sistema de **gestión de dependencias (librerías)** con descarga automática
- Sistema de **construcción de proyectos** (de código a paquete entregable y ejecutable)
- Estructura única de proyecto compatible con todos los **entornos de desarrollo** y sistemas de CI

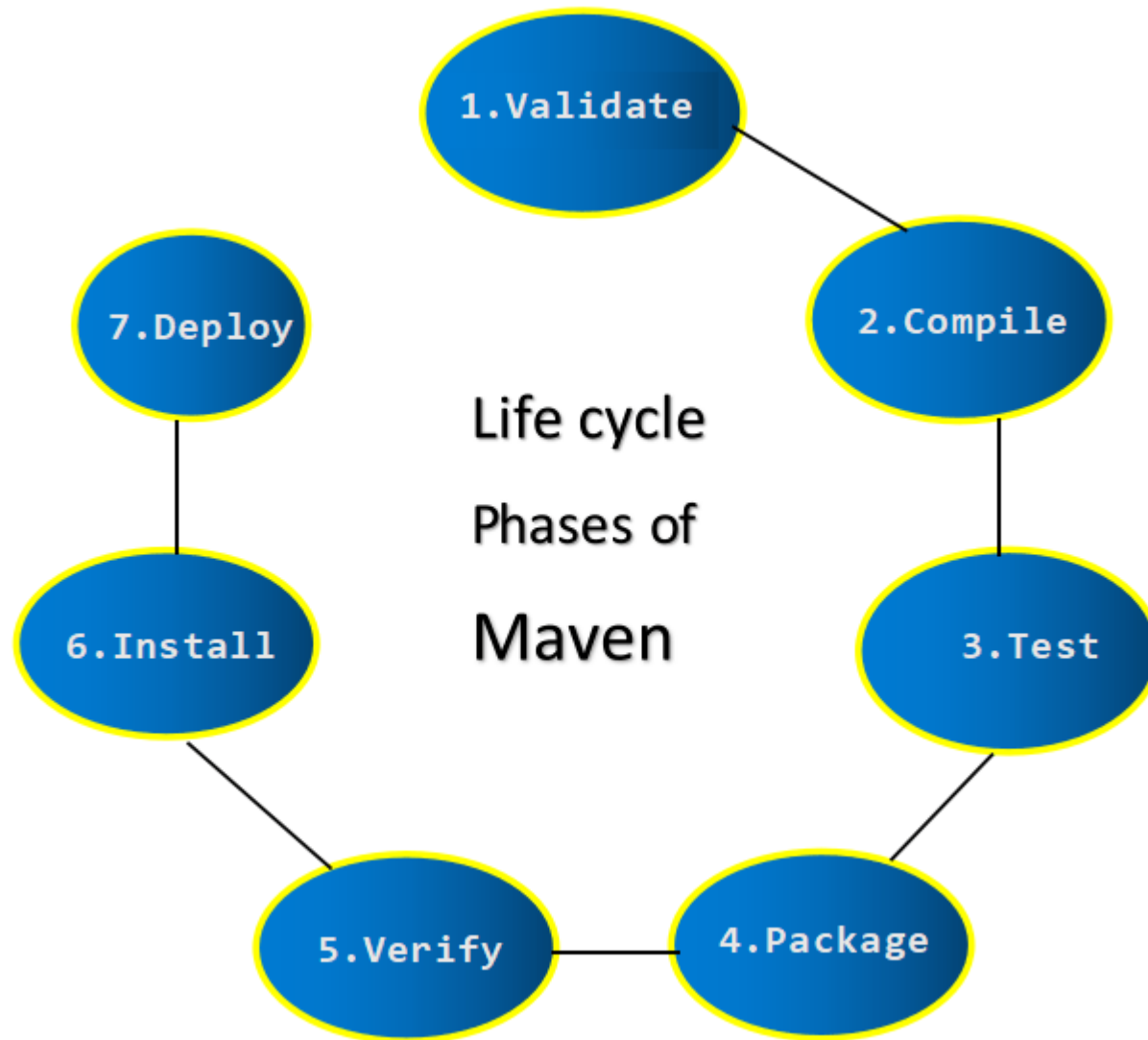


Maven

¿Qué es Maven?

- Maven gestiona muchos aspectos de la **construcción** de la aplicación
 - Descarga automática de dependencias
 - Compilación
 - Ejecución de tests
 - Publicación de la aplicación construida (binarios)
 - Documentación

Maven



Maven

Life-cycle de Maven

- Es un proceso de construcción y distribución de artefactos software (proyectos)
- Cuenta con una serie de pasos secuenciales
- Si uno de los pasos falla, no se continúa con el siguiente
- Maven cuenta con diferentes plugins que permiten llevar a cabo estas pasos de manera automática
- Podemos ejecutar solo hasta cierta fase (p.e. Maven Test)

Life-cycle de Maven

- **Validate:** Comprueba que el proyecto es correcto y tiene toda la información necesaria
- **Compile:** Compila el código fuente de la aplicación
- **Test:** Compila el código de los test y los ejecuta
- **Package:** Empaqueta el código fuente en un formato distribuible (p.e. JAR)
- **Verify:** Comprueba checks de la aplicación
- **Install:** Instala el paquete en el repositorio local (.m2)
- **Deploy:** Copia el paquete final a un repositorio remoto (p.e. Maven Central)

Maven

¿Cómo se usa Maven?

- **Desde el entorno de desarrollo:** Maven está integrado en los IDEs Java más importantes (Eclipse, IntelliJ, Visual Studio Code...)
 - Instalado por defecto junto al IDE
- **Desde la línea de comandos:** Sin necesidad de usar un IDE. Ideal para construcción de proyectos de forma automatizada
 - Debe instalarse en la máquina

Maven

Maven desde línea de comandos

- Es necesario instalar el binario en nuestra máquina o en la máquina de CI
- Para **Java 17** es necesario instalar **Maven 3.8**
- Descarga e instalación
 - Oficial: <https://maven.apache.org/download.cgi>
 - Windows (Guía):
<https://mkyong.com/maven/how-to-install-maven-in-windows/>
 - Linux (Guía):
<https://computingforgeeks.com/install-latest-apache-maven-on-ubuntu/>

Maven

Maven desde línea de comandos

- Una vez instalado, podemos comprobar su versión

```

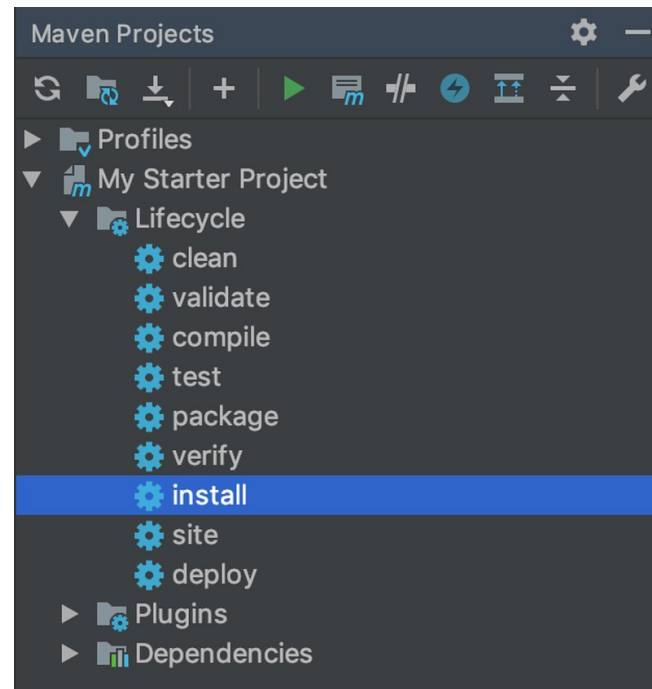
michel@maes-msi: ~
michel@maes-msi: ~ 80x9
michel@maes-msi:~$ mvn -v
Apache Maven 3.8.4 (9b656c72d54e5bacbed989b64718c159fe39b537)
Maven home: /home/michel/Desarrollo/apache-maven-3.8.4
Java version: 17.0.2, vendor: Private Build, runtime: /usr/lib/jvm/java-17-openj
dk-amd64
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.4.0-104-generic", arch: "amd64", family: "unix"
michel@maes-msi:~$

```

Maven desde línea de comandos

- Desde la herramienta de comandos podemos ejecutar todas las acciones que realizamos de manera interactiva desde el IDE

- mvn clean
- mvn validate
- mvn compile
- mvn test
- mvn package
- ...



Maven

mvn test

- Ejecuta todos los test del framework de testing que hayamos configurado (p.e. JUnit)

mvn test

```

-----
T E S T S
-----
Running es.codeurjc.test.ejem.ListTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.076 sec - in es.codeurjc.test.ejem.ListTest
Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.519 s
[INFO] Finished at: 2022-03-16T15:49:09+01:00

```

Maven

mvn test

- Es posible filtrar los tests y ejecutar únicamente una clase o método

```
mvn test -Dtest=ClassName#Method
```

```
mvn test -Dtest=AppTest
```

```
mvn test -Dtest=Calculadora3Test#testResta
```

Maven

mvn package

- Empaqueta el proyecto para su distribución
- Es común generar un JAR (en la carpeta target)
 - Incluye la aplicación con todas sus dependencias
 - No incluye los test
 - Para aplicaciones web o interactivas, puede ser ejecutado directamente teniendo Java instalado
- Requiere de configuración adicional en el pom.xml en función del proyecto

Maven

mvn package con SpringBoot

- Usaremos el proyecto **selenium_ejem2**
- SpringBoot cuenta con un plugin para generar un JAR ejecutable de manera sencilla

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

Maven

mvn package con SpringBoot

- Usaremos el proyecto **selenium_ejem2**
- Construimos el JAR ejecutable
 - mvn package
- Ejecutamos el JAR como una aplicación
 - java -jar target/selenium_ejem2-0.0.1-SNAPSHOT.jar

TIP: En caso de no querer ejecutar los test, es posible “ignorarlos” con **mvn package -DskipTests**