

Lab 02 CVDS

Miguel Monroy

Santiago Rocha

Apache Maven

¿Cuál es su mayor utilidad?

Maven se utiliza en la gestión y construcción de software. Posee la capacidad de realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado. Es decir, hace posible la creación de software con dependencias incluidas dentro de la estructura del JAR.

¿Fases de maven?

- validate
- initialize
- generate-sources
- process-sources
- generate-resources
- process-resources
- compile
- process-classes
- generate-test-sources
- process-test-sources
- generate-test-resources
- process-test-resources
- test-compile
- process-test-classes
- test
- prepare-package
- package
- pre-integration-test
- integration-test
- post-integration-test
- verify
- install
- deploy

Ciclo de vida de la construcción?

Las partes del ciclo de vida principal del proyecto Maven son:

Compilar: Genera los ficheros .class compilando los fuentes .java

Tests: Ejecuta los test automáticos de JUnit existentes, abortando el proceso si alguno de ellos falla.

Packages: Genera el fichero .jar con los .class compilados

Instalar: Copia el fichero .jar a un directorio de nuestro ordenador donde maven deja todos los .jar. De esta forma esos .jar pueden utilizarse en otros proyectos maven en el mismo ordenador.

Desplegar: Copia el fichero .jar a un servidor remoto, poniéndolo disponible para cualquier proyecto maven con acceso a ese servidor remoto.

¿Para qué sirven los plugins?

En Maven, los plugins tienen metas, que según el técnico son solo métodos Java. Las metas ejecutan tareas de construcción como: compilar el proyecto, empaquetarlo e implementarlo en un servidor local o remoto. Esas actividades se correlacionan perfectamente para construir fases del ciclo de vida.

¿Qué es y para qué sirve el repositorio central de maven?

El repositorio central de Maven es la ubicación predeterminada para que Maven descargue todas las bibliotecas de dependencia del proyecto para su uso. Para cualquier biblioteca involucrada en el proyecto, Maven primero busca en la carpeta .m2 del Repositorio local, y si no encuentra la biblioteca necesaria, busca en el Repositorio central y carga la biblioteca en el repositorio local.

EJERCICIO DE LAS FIGURAS CREAR UN PROYECTO CON MAVEN

Buscar cómo se crea un proyecto maven con ayuda de los arquetipos (archetypes).

Busque cómo ejecutar desde línea de comandos el objetivo "generate" del plugin

"archetype", con los siguientes parámetros:

Grupo: edu.eci.cvds

Id del Artefacto: Patterns

Paquete: edu.eci.cvds.patterns

archetypeArtifactId: maven-archetype-quickstart

```
mvn archetype:generate DarchetypeArtifactId: maven-archetype-quickstart
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1865]
(c) Microsoft Corporation. All rights reserved.

C:\Users\santiago.rocha-c\Desktop\Lab02_Monroy_Rocha>mvn archetype:generate
[INFO] Scanning for projects...
[INFO] -----< org.apache.maven:standalone-pom >-----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----[ pom ]-----
[INFO]
[INFO] >>> maven-archetype-plugin:3.2.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO] <<< maven-archetype-plugin:3.2.1:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO] --- maven-archetype-plugin:3.2.1:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Interactive mode
[INFO] No archetype defined. Using maven-archetype-quickstart (org.apache.maven.archetypes:maven-archetype-quickstart:1.0)
Choose archetype:
1: remote -> am.ik.archetype:elm-spring-boot-blank-archetype (Blank multi project for Spring Boot + Elm)
2: remote -> am.ik.archetype:graalvm-blank-archetype (Blank project for GraalVM)
3: remote -> am.ik.archetype:graalvm-springmvc-blank-archetype (Blank project for GraalVM + Spring MVC)
4: remote -> am.ik.archetype:graalvm-springwebflux-blank-archetype (Blank project for GraalVM + Spring MVC)
5: remote -> am.ik.archetype:maven-reactjs-blank-archetype (Blank Project for React.js)
6: remote -> am.ik.archetype:msgpack-rpc-jersey-blank-archetype (Blank Project for Spring Boot + Jersey)
7: remote -> am.ik.archetype:mvc-1.0-blank-archetype (MVC 1.0 Blank Project)
8: remote -> am.ik.archetype:spring-boot-blank-archetype (Blank Project for Spring Boot)
9: remote -> am.ik.archetype:spring-boot-docker-blank-archetype (Docker Blank Project for Spring Boot)
10: remote -> am.ik.archetype:spring-boot-gae-blank-archetype (GAE Blank Project for Spring Boot)
```

```
C:\Windows\System32\cmd.exe
B136: remote -> za.co.absa.hyperdrive:component-archetype_2.12 (-)
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains): 1942:
Choose org.apache.maven.archetypes:maven-archetype-quickstart version:
1: 1.0-alpha-1
2: 1.0-alpha-2
3: 1.0-alpha-3
4: 1.0-alpha-4
5: 1.0
6: 1.1
7: 1.3
8: 1.4
Choose a number: 8: 8
Define value for property 'groupId': edu.eci.cvds
Define value for property 'artifactId': Patterns
Define value for property 'version' 1.0-SNAPSHOT: :
Define value for property 'package' edu.eci.cvds: :
Confirm properties configuration:
groupId: edu.eci.cvds
artifactId: Patterns
version: 1.0-SNAPSHOT
package: edu.eci.cvds
Y: : Y
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.4
[INFO] -----
[INFO] Parameter: groupId, Value: edu.eci.cvds
[INFO] Parameter: artifactId, Value: Patterns
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: edu.eci.cvds
[INFO] Parameter: packageInPathFormat, Value: edu/eci/cvds
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 04:44 min
[INFO] Finished at: 2022-08-18T16:58:26-05:00
[INFO] -----

C:\Users\santiago.rocha-c\Desktop\Lab02_Monroy_Rocha>
```

```
C:\Windows\System32\cmd.exe

C:\Users\santiago.rocha-c\Desktop\Lab02_Monroy_Rocha\Patterns>tree /f
Folder PATH listing
Volume serial number is F671-CE19
C:.
  pom.xml
  src
    main
      java
        edu
          eci
            cvds
              App.java
    test
      java
        edu
          eci
            cvds
              AppTest.java

C:\Users\santiago.rocha-c\Desktop\Lab02_Monroy_Rocha\Patterns>
```

AJUSTAR ALGUNAS CONFIGURACIONES EN EL PROYECTO

```
File Edit Selection View Go Run Terminal Help pom.xml - Visual Studio Code

pom.xml X
C:\> Users > santiago.rocha-c > Desktop > Lab02_Monroy_Rocha > Patterns > pom.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5    <modelVersion>4.0.0</modelVersion>
6
7    <groupId>edu.eci.cvds</groupId>
8    <artifactId>Patterns</artifactId>
9    <version>1.0-SNAPSHOT</version>
10
11    <name>Patterns</name>
12    <!-- FIXME change it to the project's website -->
13    <url>http://www.example.com</url>
14
15    <properties>
16      <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
17      <maven.compiler.source>1.8</maven.compiler.source>
18      <maven.compiler.target>1.8</maven.compiler.target>
19    </properties>
20
21    <dependencies>
22      <dependency>
23        <groupId>junit</groupId>
24        <artifactId>junit</artifactId>
25        <version>4.11</version>
26        <scope>test</scope>
27      </dependency>
28    </dependencies>
29
30    <build>
31      <pluginManagement><!-- lock down plugins versions to avoid -->
32        <plugins>
33          <!-- clean lifecycle, see https://maven.apache.org
34          <plugin>
35            <artifactId>maven-clean-plugin</artifactId>
```

```
C:\Windows\System32\cmd.exe
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-archiver/3.1.1/maven-archiver-3.1.1.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/3.4/plexus-archiver-3.4.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/3.0.1/maven-shared-utils-3.0.1.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-compress/1.11/commons-compress-1.11.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/2.7.1/plexus-io-2.7.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-archiver/3.1.1/maven-archiver-3.1.1.jar (24 kB at 152 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/tukaani/xz/1.5/xz-1.5.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/2.7.1/plexus-io-2.7.1.jar (86 kB at 343 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/3.4/plexus-archiver-3.4.jar (187 kB at 748 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/tukaani/xz/1.5/xz-1.5.jar (100 kB at 289 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/3.0.1/maven-shared-utils-3.0.1.jar (154 kB at 230 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-compress/1.11/commons-compress-1.11.jar (426 kB at 383 kB/s)
[INFO] Building jar: C:\Users\santiago.rocha-c\Desktop\Lab02_Monroy_Rocha\Patterns\target\Patterns-1.0-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 28.873 s
[INFO] Finished at: 2022-08-18T17:10:40-05:00
[INFO] -----
C:\Users\santiago.rocha-c\Desktop\Lab02_Monroy_Rocha\Patterns>
```

Busque cuál es el objetivo del parámetro "package" y qué otros parámetros se podrían enviar al comando mvn.

mvn package: Es el comando de empaquetado del proyecto maven. Para el proyecto java, ejecute el paquete y márkelo como un paquete jar, y para el proyecto web es un paquete war.

Busque cómo ejecutar desde línea de comandos, un proyecto maven y verifique la salida cuando se ejecuta con la clase App.java como parámetro en "mainClass".

En el POM, modificar:

```

<!-- To include exec plugin -->
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>exec-maven-plugin</artifactId>
  <version>3.1.0</version>
  <executions>
    <execution>
      <goals>
        <goal>java</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <mainClass>edu.eci.cvds.App</mainClass>
    <arguments>
      <argument>argument1</argument>
    </arguments>
    <systemProperties>
      <systemProperty>
        <key>myproperty</key>
        <value>myvalue</value>
      </systemProperty>
    </systemProperties>
  </configuration>
</plugin>

```

Luego, mvn package, y por último:

```

C:\Users\santiago.rocha-c\Desktop\Lab02_Monroy_Rocha\Patterns>mvn exec:java -Dexec.mainClass="App"
[INFO] Scanning for projects...
[INFO]
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:3.1.0:java (default-cli) @ Patterns ---
Hello World!
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 0.468 s
[INFO] Finished at: 2022-08-18T17:58:28-05:00
[INFO]
C:\Users\santiago.rocha-c\Desktop\Lab02_Monroy_Rocha\Patterns>

```

Buscar cómo enviar parámetros al plugin "exec".

-Dexec.args="argumentos_necesarios"

Ejecutar nuevamente la clase desde línea de comandos y verificar la salida: Hello World!

```

[INFO] --- exec-maven-plug
Hello World!
[INFO] -----

```

Ejecutar la clase desde línea de comandos enviando su nombre como parámetro y verificar la salida. Ej: Hello Pepito!

```
C:\Users\santiago.rocha-c\Desktop\Lab02_Monroy_Rocha\Patterns>mvn exec:java -Dexec.mainClass="App" -Dexec.args="Miguel"
[INFO] Scanning for projects...
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- exec-maven-plugin:3.1.0:java (default-cli) @ Patterns ---
Hello Miguel!
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.468 s
[INFO] Finished at: 2022-08-18T18:15:31-05:00
[INFO] -----
```

Ejecutar la clase con su nombre y apellido como parámetro. ¿Qué sucedió?

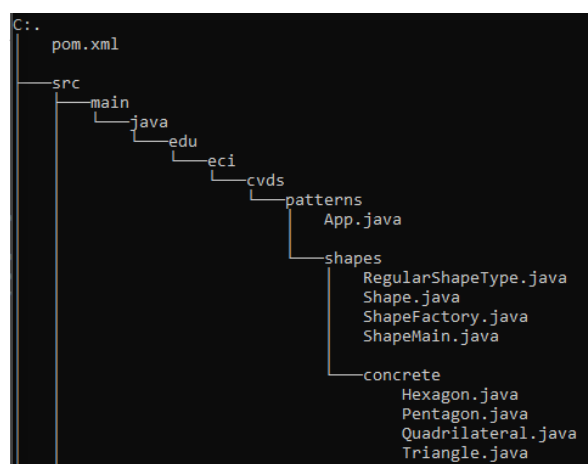
```
C:\Users\santiago.rocha-c\Desktop\Lab02_Monroy_Rocha\Patterns>mvn exec:java -Dexec.mainClass="App" -Dexec.args="Miguel Monroy"
[INFO] Scanning for projects...
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- exec-maven-plugin:3.1.0:java (default-cli) @ Patterns ---
Hello Miguel Monroy!
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.453 s
[INFO] Finished at: 2022-08-18T18:16:41-05:00
[INFO] -----
```

Verifique cómo enviar los parámetros de forma "compuesta" para que el saludo se realice con nombre y apellido.

Ejecutar nuevamente y verificar la salida en consola. Ej: Hello Pepito Perez!

```
C:\Users\santiago.rocha-c\Desktop\Lab02_Monroy_Rocha\Patterns>mvn exec:java -Dexec.mainClass="App" -Dexec.args="'Miguel' 'Monroy'"
[INFO] Scanning for projects...
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- exec-maven-plugin:3.1.0:java (default-cli) @ Patterns ---
Hello Miguel Monroy!
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.453 s
[INFO] Finished at: 2022-08-18T18:20:07-05:00
[INFO] -----
```

Cree el archivo ShapeFactory.java en el directorio src/main/java/edu/eci/cvds/patterns/shapes implementando el patrón fábrica, haciendo uso de la instrucción switch-case de Java y usando las enumeraciones.



Ejecute múltiples veces la clase ShapeMain, usando el plugin exec de maven con los siguientes parámetros y verifique la salida en consola para cada una:

Sin parametros

```
C:\Users\miguel.monroy-c\Downloads\Patterns>mvn exec:java
[INFO] Scanning for projects...
[INFO]
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:1.6.0:java (default-cli) @ Patterns ---
Parameter of type RegularShapeType is required.
```

- "qwerty"

```
C:\Users\miguel.monroy-c\Downloads\Patterns>mvn exec:java -Dexec.args="qwerty"
[INFO] Scanning for projects...
[INFO]
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:1.6.0:java (default-cli) @ Patterns ---
Parameter of type RegularShapeType is required.
```

- "Pentagon"

```
C:\Users\miguel.monroy-c\Downloads\Patterns>mvn exec:java -Dexec.args="Pentagon"
[INFO] Scanning for projects...
[INFO]
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:1.6.0:java (default-cli) @ Patterns ---
Successfully created a Pentagon with 5 sides.
```

- "Hexagon"

```
C:\Users\miguel.monroy-c\Downloads\Patterns>mvn exec:java -Dexec.args="Hexagon"
[INFO] Scanning for projects...
[INFO]
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:1.6.0:java (default-cli) @ Patterns ---
Successfully created a Hexagon with 6 sides.
```

¿Cuál(es) de las anteriores instrucciones se ejecutan y funcionan correctamente y por qué?

Todas las instrucciones anteriormente utilizadas funcionan correctamente debido a que para caso de prueba se obtiene una salida que concuerda con lo esperado.