

LABORATORIO 2- PATTERNS

JUAN DAVID MARTINEZ CARDOZO

JHON SEBASTIAN PIÑEROS

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

CICLOS DE VIDA DEL DESARROLLO DE SOFTWARE

OCTAVO SEMESTRE

FEBRERO, 2022

BOGOTA D.C

# LA HERRAMIENTA MAVEN

- Cuál es su mayor utilidad

RTA: La mayor utilidad de Maven es manejar proyectos, automatizando el pipeline para nuestros trabajos

- Fases de maven

Clean Lifecycle	Default Lifecycle		Site Lifecycle
pre-clean	validate	test-compile	pre-site
<b>clean</b>	initialize	process-test-classes	<b>site</b>
post-clean	generate-sources	test	post-site
	process-sources	prepare-package	site-deploy
	generate-resources	<b>package</b>	
	process-resources	pre-integration-test	
	<b>compile</b>	integration-test	
	process-classes	post-integration-test	
	generate-test-sources	verify	
	process-test-sources	<b>install</b>	
	generate-test-resources	<b>deploy</b>	
	process-test-resources		

- Ciclo de vida de la construcción

RTA: Tiene 3 ciclos, clean, default y site

- Para qué sirven los plugins

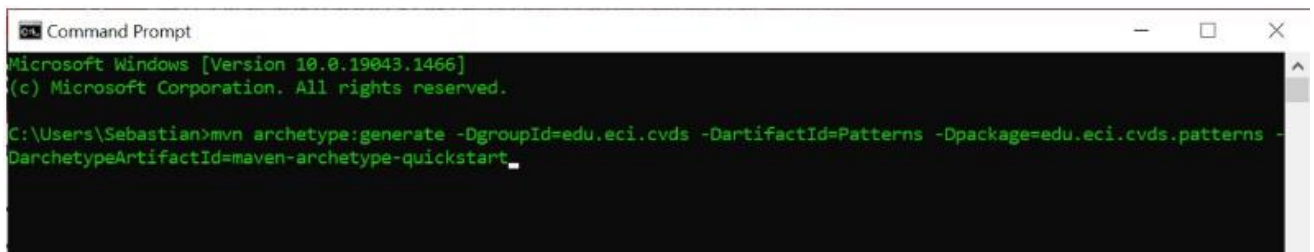
RTA: Los plugins sirven para añadir funcionalidades extra o mejoras a el programa

- Qué es y para qué sirve el repositorio central de Maven

RTA: Acá se encuentran todas las librerías que podemos usar en Maven

# 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



```
Microsoft Windows [Version 10.0.19043.1466]
(c) Microsoft Corporation. All rights reserved.

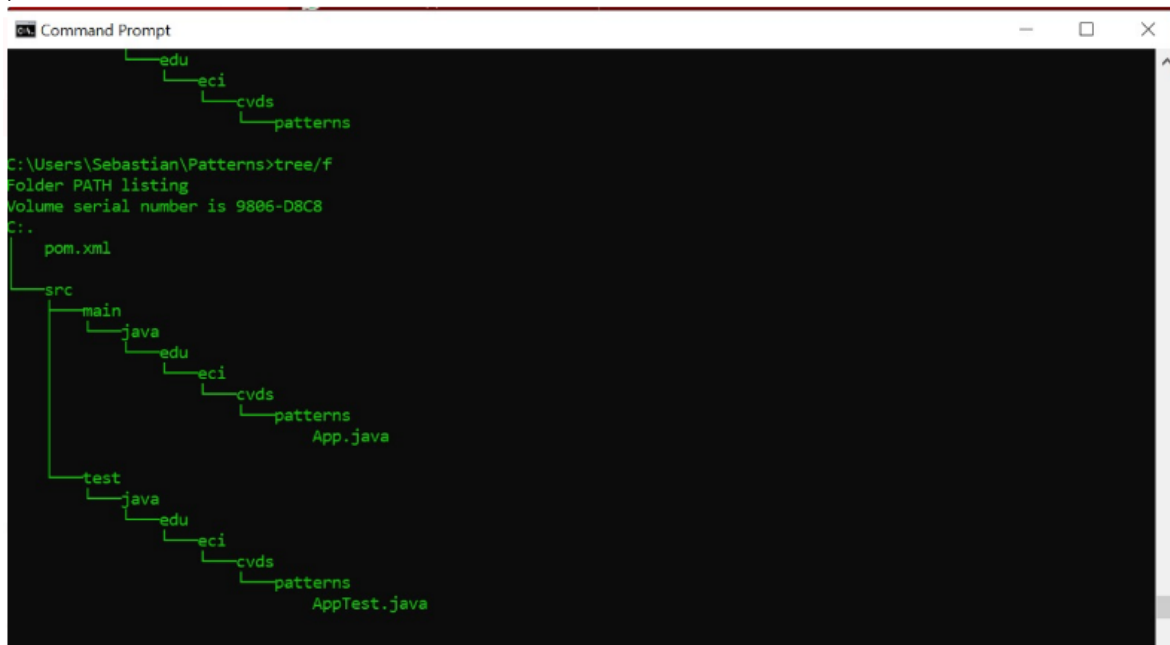
C:\Users\Sebastian>mvn archetype:generate -DgroupId=edu.eci.cvds -DartifactId=Patterns -Dpackage=edu.eci.cvds.patterns -DarchetypeArtifactId=maven-archetype-quickstart
```

Cambie al directorio **Patterns**:

```
$ cd Patterns
```

Para ver el conjunto de archivos y directorios creados por el comando **mvn** ejecute el comando **tree**.

```
$ tree
```



```
Microsoft Windows [Version 10.0.19043.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sebastian\Patterns>tree /f
Folder PATH listing
Volume serial number is 9806-D8C8
C:..
  pom.xml
  src
    main
      java
        edu
          eci
            cvds
              patterns
                App.java
    test
      java
        edu
          eci
            cvds
              patterns
                AppTest.java
```

# AJUSTAR ALGUNAS CONFIGURACIONES EN EL PROYECTO

- Edite el archivo `pom.xml` y realice la siguiente actualización:
- Hay que cambiar la version del compilador de Java a la versión 8, para ello, agregue la sección `properties` antes de la sección de dependencias:

```
<properties>
  <maven.compiler.target>1.8</maven.compiler.target>
  <maven.compiler.source>1.8</maven.compiler.source>
</properties>
```

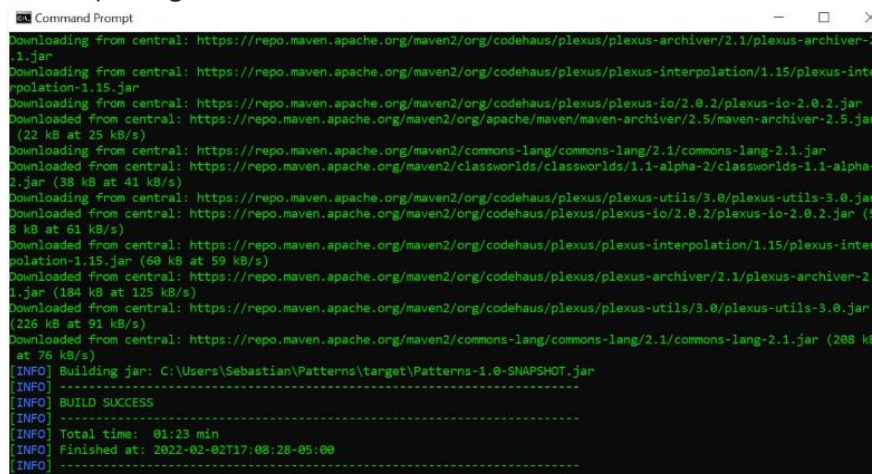


```
pom.xml: Bloc de notes
Archivo Editar Formato Vista Ayuda
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>edu.eci.cvds</groupId>
  <artifactId>Patterns</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>Patterns</name>
  <url>http://maven.apache.org</url>
  <properties>
    <maven.compiler.target>1.8</maven.compiler.target>
    <maven.compiler.source>1.8</maven.compiler.source>
  </properties>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

## COMPILAR Y EJECUTAR

Para compilar ejecute el comando:

`$ mvn package`



```
Command Prompt
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/2.1/plexus-archiver-2.1.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-interpolation/1.15/plexus-interpolation-1.15.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/2.0.2/plexus-io-2.0.2.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-archiver/2.5/maven-archiver-2.5.jar (22 kB at 25 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/commons-lang/commons-lang/2.1/commons-lang-2.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/classworlds/classworlds/1.1-alpha-2/classworlds-1.1-alpha-2.jar (38 kB at 41 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/2.0.2/plexus-io-2.0.2.jar (22 kB at 61 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-interpolation/1.15/plexus-interpolation-1.15.jar (60 kB at 59 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/2.1/plexus-archiver-2.1.jar (184 kB at 125 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.jar (226 kB at 91 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/commons-lang/commons-lang/2.1/commons-lang-2.1.jar (208 kB at 76 kB/s)
[INFO] Building jar: C:\Users\Sebastian\Patterns\target\Patterns-1.0-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:23 min
[INFO] Finished at: 2022-02-02T17:08:28-05:00
[INFO] -----
```

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

El objetivo del parámetro package es empaquetar el código compilado como un fichero tipo .jar, otros parámetros que se podrían enviar al comando mvn son: *mvn test* que ejecuta una prueba, *mvn clean* que borra elementos generados

- 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". Tip: <https://www.mojohaus.org/exec-maven-plugin/usage.html>

```
at 76 kB/s)
[INFO] Building jar: C:\Users\Sebastian\Patterns\target\Patterns-1.0-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:23 min
[INFO] Finished at: 2022-02-02T17:08:28-05:00
[INFO] -----
C:\Users\Sebastian\Patterns>mvn exec:java -Dexec.mainClass="edu.eci.cvds.patterns.App"
```

- Realice el cambio en la clase **App.java** para crear un saludo personalizado, basado en los parámetros de entrada a la aplicación. Utilizar la primera posición del parámetro que llega al método "main" para realizar el saludo personalizado, en caso que no sea posible, se debe mantener el saludo como se encuentra actualmente:
- Ejecutar nuevamente la clase desde línea de comandos y verificar la salida: **Hello World!**

```
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/3.1.0/maven-shared-utils-3.1.0.jar (164 kB at 24 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/commons-codec/commons-codec/1.11/commons-codec-1.11.jar (335 kB at 48 kB/s)
Hello World!
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 18.529 s
[INFO] Finished at: 2022-02-02T17:14:20-05:00
[INFO] -----
```

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

```
Microsoft Windows [Version 10.0.19043.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sebastian\Patterns>mvn exec:java -Dexec.mainClass="edu.eci.cvds.patterns.App" -Dexec.args="Pepito"
[INFO] Scanning for projects...
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ Patterns ---
Hello Pepito!
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.235 s
[INFO] Finished at: 2022-02-02T17:55:37-05:00
[INFO] -----

C:\Users\Sebastian\Patterns>
```

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

```
C:\Users\Sebastian\Patterns>mvn exec:java -Dexec.mainClass="edu.eci.cvds.patterns.App" -Dexec.args="Sebastian"
[INFO] Scanning for projects...
[INFO]
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ Patterns ---
Hello Sebastian!
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 1.278 s
[INFO] Finished at: 2022-02-02T17:57:23-05:00
[INFO]
C:\Users\Sebastian\Patterns>mvn exec:java -Dexec.mainClass="edu.eci.cvds.patterns.App" -Dexec.args="Barrera"
[INFO] Scanning for projects...
[INFO]
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ Patterns ---
Hello Barrera!
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 1.097 s
[INFO] Finished at: 2022-02-02T17:57:34-05:00
[INFO]
```

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

```
C:\Users\Sebastian\Patterns>mvn exec:java -Dexec.mainClass="edu.eci.cvds.patterns.App" -Dexec.args="Sebastian Barrera"
[INFO] Scanning for projects...
[INFO]
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ Patterns ---
Hello Sebastian Barrera!
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 1.135 s
[INFO] Finished at: 2022-02-02T18:00:51-05:00
[INFO]
```

- Ejecutar nuevamente y verificar la salida en consola.

```
C:\Users\Sebastian\Patterns>mvn exec:java -Dexec.mainClass="edu.eci.cvds.patterns.App" -Dexec.args="Pepito Perez"
[INFO] Scanning for projects...
[INFO]
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ Patterns ---
Hello Pepito Perez!
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 1.193 s
[INFO] Finished at: 2022-02-02T19:25:33-05:00
[INFO]
```

# HACER EL ESQUELETO DE LA APLICACIÓN

- Cree el paquete `edu.eci.cvds.patterns.shapes` y el paquete `edu.eci.cvds.patterns.shapes.concrete`.
- Cree una interfaz llamada `Shape.java` en el directorio `src/main/java/edu/eci/cvds/patterns/shapes` de la siguiente manera:

```
package edu.eci.cvds.patterns.shapes;

public interface Shape {
    public int getNumberOfEdges();
}
```

- Cree una enumeración llamada `RegularShapeType.java` en el directorio `src/main/java/edu/eci/cvds/patterns/shapes` así:

```
package edu.eci.cvds.patterns.shapes;

public enum RegularShapeType {
    Triangle, Quadrilateral, Pentagon, Hexagon
}
```

- En el directorio `src/main/java/edu/eci/cvds/patterns/shapes/concrete` cree las diferentes clases (Triangle, Quadrilateral, Pentagon, Hexagon), que implementen la interfaz creada y retornen el número correspondiente de vértices que tiene la figura. Siguiendo el ejemplo del triángulo:

```
package edu.eci.cvds.patterns.shapes.concrete;

import edu.eci.cvds.patterns.shapes.Shape;

public class Triangle implements Shape {
    public int getNumberOfEdges() {
        return 3;
    }
}
```

- Cree el archivo `ShapeMain.java` en el directorio `src/main/java/edu/eci/cvds/patterns/shapes` con el método `main`:

```
package edu.eci.cvds.patterns.shapes;

public class ShapeMain {
    public static void main(String[] args) {
        if (args == null || args.length != 1) {
            System.err.println("Parameter of type RegularShapeType is required.");
            return;
        }
        try {

            RegularShapeType type = RegularShapeType.valueOf(args[0]);
            Shape shape = ShapeFactory.create(type);

            System.out.println(String.format("Successfully created a %s with %s sides.", type, shape.getNumberOfEdges()));

        } catch (IllegalArgumentException ex) {
            System.err.println("Parameter '" + args[0] + "' is not a valid RegularShapeType");
            return;
        }
    }
}
```



- Analice y asegúrese de entender cada una de las instrucciones que se encuentran en todas las clases que se crearon anteriormente.
- 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 parámetros

```
C:\Users\Sebastian\Patterns>mvn exec:java -Dexec.mainClass="edu.eci.cvds.patterns.shapes.ShapeMain"
[INFO] Scanning for projects...
[INFO]
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ Patterns ---
Parameter of type RegularShapeType is required.
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 1.145 s
[INFO] Finished at: 2022-02-02T19:28:59-05:00
[INFO]
C:\Users\Sebastian\Patterns>
```

- Parámetro: `qwerty`

```
C:\Users\Sebastian\Patterns>mvn exec:java -Dexec.mainClass="edu.eci.cvds.patterns.shapes.ShapeMain" -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:3.0.0:java (default-cli) @ Patterns ---
Parameter 'qwerty' is not a valid RegularShapeType
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 1.118 s
[INFO] Finished at: 2022-02-02T19:31:43-05:00
[INFO]
C:\Users\Sebastian\Patterns>
```

- Parámetro: `pentagon`

```
C:\Users\Sebastian\Patterns>mvn exec:java -Dexec.mainClass="edu.eci.cvds.patterns.shapes.ShapeMain" -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:3.0.0:java (default-cli) @ Patterns ---
Parameter 'pentagon' is not a valid RegularShapeType
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 1.159 s
[INFO] Finished at: 2022-02-02T19:32:15-05:00
[INFO]
C:\Users\Sebastian\Patterns>
```



- Parámetro Hexagon

```
C:\Users\Sebastian\Patterns>mvn exec:java -Dexec.mainClass="edu.eci.cvds.patterns.shapes.ShapeMain" -Dexec.args="Hexagon"
[INFO] Scanning for projects...
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ Patterns ---
Successfully created a Hexagon with 6 sides.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.247 s
[INFO] Finished at: 2022-02-02T19:32:59-05:00
[INFO] -----
```

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

RTA: Todas las anteriores instrucciones se ejecutan, pero no funcionan ya que no están definidas, solo funciona la última de hexagon ya que si esta definida