

CREAR UN PROYECTO CON MAVEN

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

- Creamos una carpeta para el laboratorio y realizar el proyecto de maven

```
C:\Users\juan.rrodriguez.LABINFO>mkdir Lab2  
C:\Users\juan.rrodriguez.LABINFO>cd Lab2
```

- El comando `mvn archetype:generate` genera un proyecto de maven

```
PS C:\Users\esteban.aguilera-c.LABINFO\lab2> mvn archetype:generate
```

2. Busque cómo ejecutar desde línea de comandos el objetivo "generate" del plugin "archetype", con los siguientes parámetros:

ProjectId: `org.apache.maven.archetypes:maven-archetype-quickstart:1.0`
Id *del* *Grupo:* `edu.eci.cvds`
Id *del* *Artefacto:* `Patterns`
Paquete: `edu.eci.cvds.patterns.archetype`

- Luego de ejecutar `mvn archetype:generate` nos pide llenar el id, artefacto , paquete y version. Ejecutamos en la consola todos estos requerimientos

```
Choose a number of applicable patterns (format: [groupId]:[artifactId], case-sensitive contains):  
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  
9: 1.5  
Choose a number: 9: 5
```

```
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: edu.eci.cvds.patterns.archetype  
Confirm properties configuration:  
groupId: edu.eci.cvds  
artifactId: Patterns  
version: 1.0-SNAPSHOT  
package: edu.eci.cvds.patterns.archetype  
Y: Y
```

3. Cambie al directorio Patterns:

- a. Para cambiar el directorio usamos cd

```
PS C:\Users\esteban.aguilera-c.LABINFO\Documents\lab> cd patterns
```

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

```
PS C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns> tree
Folder PATH listing
Volume serial number is 9437-6886
C:.\
├── src
│   ├── main
│   │   ├── java
│   │   │   ├── edu
│   │   │   │   ├── eci
│   │   │   │   │   ├── cvds
│   │   │   │   │   │   ├── patterns
│   │   │   │   │   │   │   └── archetype
│   │   └── test
│   │       ├── java
│   │       │   ├── edu
│   │       │   │   ├── eci
│   │       │   │   │   ├── cvds
│   │       │   │   │   │   ├── patterns
│   │       │   │   │   │   │   └── archetype
└──
```

AJUSTAR ALGUNAS CONFIGURACIONES EN EL PROYECTO

1. 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>
```

```
C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns > pom.xml
1  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
3    <modelVersion>4.0.0</modelVersion>
4    <groupId>edu.eci.cvds</groupId>
5    <artifactId>Patterns</artifactId>
6    <packaging>jar</packaging>
7    <version>1.0-SNAPSHOT</version>
8    <name>Patterns</name>
9    <url>http://maven.apache.org</url>
10   <properties>
11     <maven.compiler.target>1.8</maven.compiler.target>
12     <maven.compiler.source>1.8</maven.compiler.source>
13   </properties>
14   <dependencies>
15     <dependency>
16       <groupId>junit</groupId>
17       <artifactId>junit</artifactId>
18       <version>3.8.1</version>
19       <scope>test</scope>
20     </dependency>
21   </dependencies>
22 </project>
23
```

COMPILAR Y EJECUTAR

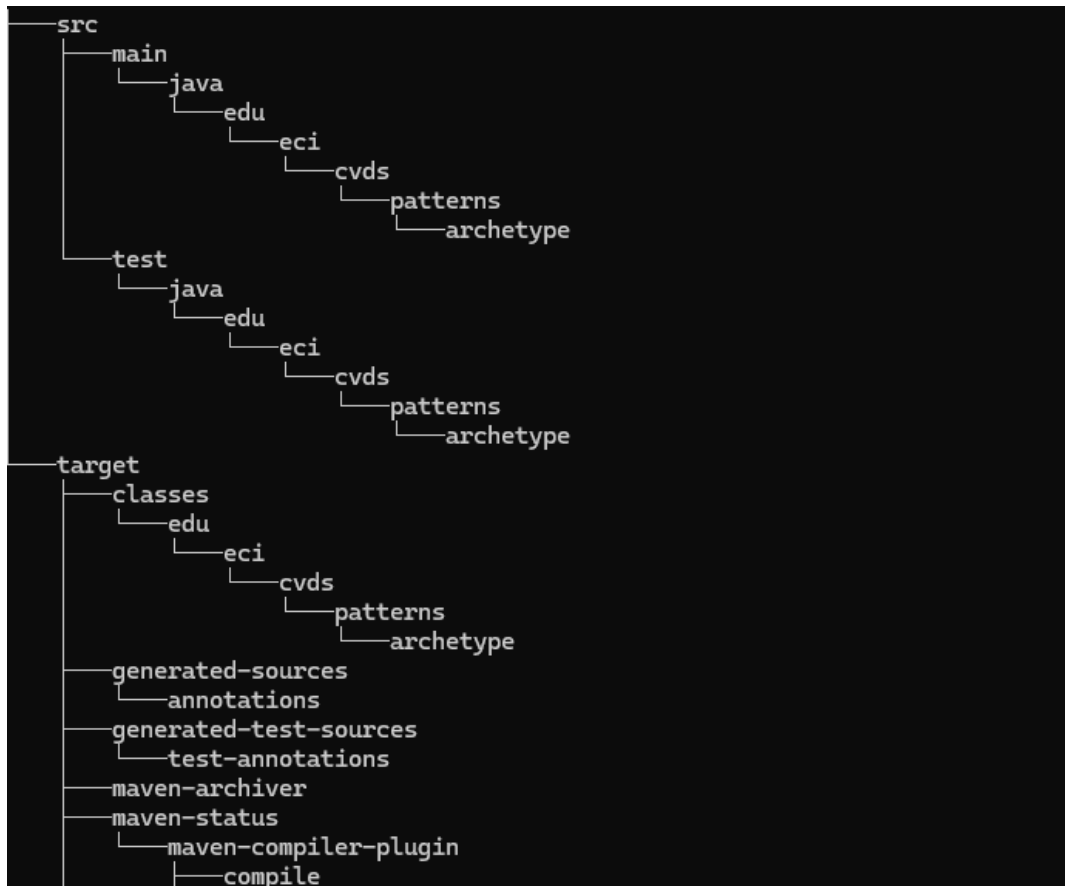
1. *Para compilar ejecute el comando:*

`$ mvn package`

Si maven no actualiza las dependencias utilice la opción -U así:

`$ mvn -U package`

```
PS C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns> mvn -U package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ Patterns ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\resources
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ Patterns ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ Patterns ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\test\resources
[INFO]
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ Patterns ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- surefire:3.2.2:test (default-test) @ Patterns ---
[INFO] Using auto detected provider org.apache.maven.surefire.junit.JUnit3Provider
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running edu.eci.cvds.patterns.archetype.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.016 s -- in edu.eci.cvds.patterns.archetype.App
```



1. Busque cuál es el objetivo del parámetro "package" y qué otros parámetros se podrían enviar al comando mvn.
 - El parámetro package en Maven es un fase del ciclo de vida de construcción que se encarga de compilar el código, ejecutar pruebas y empaquetar el proyecto en un archivo distribuable, como un .jar o .war, según la configuración del pom.xml.
 - Ahora bien cuando ejecutamos el "mvn package", Maven realiza los siguientes pasos en el ciclo de vida del proyecto:
 - **validate:** Verifica que el proyecto esté correctamente estructurado.
 - **compile:** Compila el código fuente.
 - **test:** Ejecuta las pruebas unitarias con JUnit o TestNG.
 - **package:** Empaqueta el código compilado y sus dependencias en un artefacto (como un .jar o .war)
 - Si no hay errores, generará un archivo en la carpeta target/
 - Otros parámetros útiles en Maven:

Comando	Descripción
mvn clean	Elimina los archivos generados en target/.

<code>mvn compile</code>	Compila el código fuente sin ejecutar pruebas.
<code>mvn test</code>	Ejecuta las pruebas unitarias sin empaquetar el proyecto.
<code>mvn install</code>	Compila, prueba y empaqueta el proyecto, luego instala el artefacto en el repositorio local
<code>mvn deploy</code>	Sube el artefacto a un repositorio remoto, útil para compartir librerías con otros desarrolladores.
<code>mvn verify</code>	Verifica que el paquete cumple con los requisitos (por ejemplo, ejecutando pruebas de integración).
<code>mvn site</code>	Genera documentación del proyecto en HTML.

2. 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>
 - a. Para ejecutar un proyecto Maven desde la línea de comandos, podemos usar el **mvn exec:java**, que permite ejecutar una clase main desde un proyecto Maven sin necesidad de empaquetarlo en un jar separado.
 - b. Modificamos la clase pom agregándole el plugin para compilar la clase app

```

> Users > esteban.aguilera-CLABINFO > Documents > lab > Patterns > pom.xml
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>edu.eci.cvds</groupId>
5   <artifactId>Patterns</artifactId>
6   <packaging>jar</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>Patterns</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <maven.compiler.target>1.8</maven.compiler.target>
12    <maven.compiler.source>1.8</maven.compiler.source>
13  </properties>
14  <dependencies>
15    <dependency>
16      <groupId>junit</groupId>
17      <artifactId>junit</artifactId>
18      <version>3.8.1</version>
19      <scope>test</scope>
20    </dependency>
21  </dependencies>
22  <build>
23    <plugins>
24      <plugin>
25        <groupId>org.codehaus.mojo</groupId>
26        <artifactId>exec-maven-plugin</artifactId>
27        <version>3.5.0</version>
28        <executions>
29          <execution>
30            <goals>
31              <goal>java</goal>
32            </goals>
33          </execution>
34        </executions>
35        <configuration>
36          <mainClass>edu.eci.cvds.patterns.archetype.App</mainClass>
37        </configuration>
38      </plugin>
39    </plugins>
40  </build>
41 </project>
42

```

3. Realice el cambio en la clase App.java para crear un saludo personalizado, basado en los parámetros de entrada a la aplicación.
 - a. Modificamos la clase App.java pasandole un argumento a la impresión del mensaje

```
pom.xml App.java X
C: > Users > juan.rrodriguez.LABINFO > Lab2 > Patterns > src > main > java > edu > eci > cvds > pattern
1 package edu.eci.cvds.patterns.archetype;
2
3 public class App {
4     public static void main(String[] args) {
5         if (args.length > 0) {
6             System.out.println("Hello " + args[0]);
7         } else {
8             System.out.println("Hello World!");
9         }
10    }
11 }
12
```

4. Utilizar la primera posición del parámetro que llega al método "main" para realizar el saludo personalizado, en caso de que no sea posible, se debe mantener el saludo como se encuentra actualmente:
 - a. Ya está implementado en el punto 3
5. Buscar cómo enviar parámetros al plugin "exec".
 - a. El comando para ejecutar el java con un argumento es : `mvn exec:java -Dexec.args="Juliansito"`
6. Ejecutar nuevamente la clase desde línea de comandos y verificar la salida: Hello World!

```
C:\Users\juan.rrodriguez.LABINFO\Lab2\Patterns>mvn exec:java
[INFO] Scanning for projects...
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO] --- exec:3.5.0:java (default-cli) @ Patterns ---
Hello World!
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.274 s
[INFO] Finished at: 2025-01-30T15:23:46-05:00
[INFO] -----
```

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

```
C:\Users\juan.rrodriguez.LABINFO\Lab2\Patterns>mvn exec:java -Dexec.args="Juliansito"
[INFO] Scanning for projects...
[INFO]
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec:3.5.0:java (default-cli) @ Patterns ---
Hello Juliansito!
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time:  0.266 s
[INFO] Finished at: 2025-01-30T15:15:21-05:00
[INFO]
```

8. Ejecutar la clase con su nombre y apellido como parámetro. ¿Qué sucedió?
- Envía solo el primer argumento que se haya escrito

```
C:\Users\juan.rrodriguez.LABINFO\Lab2\Patterns>mvn exec:java -Dexec.args="Juliansito rodriguez"
[INFO] Scanning for projects...
[INFO]
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec:3.5.0:java (default-cli) @ Patterns ---
Hello Juliansito
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time:  0.250 s
[INFO] Finished at: 2025-01-30T15:24:49-05:00
[INFO]
```

9. Verifique cómo enviar los parámetros de forma "compuesta" para que el saludo se realice con nombre y apellido.
- a. Agregamos un for para imprimir todos los argumentos que se manden por consola


```
pom.xml App.java X
C: > Users > esteban.aguilera-c.LABINFO > Documents > lab > Patterns > src > main > java > edu > eci > cvds > pa
1  package edu.eci.cvds.patterns.archetype;
2
3  /**
4   * Hello world!
5   *
6   */
7  public class App
8  {
9      public static void main( String[] args )
10     {
11         if(args.length > 0){
12             String message ="" ;
13
14             for(int i = 0; i < args.length; i++) {
15                 message += " " + args[i];
16             }
17             System.out.println( "Hello"+message);
18
19         }else{
20             System.out.println( "Hello World!");
21         }
22     }
23 }
24
```

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

a. Ejecutamos el programa con Esteban Aguilera

```
C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns>mvn exec:java -Dexec.args="Esteban Aguilera"
[INFO] Scanning for projects...
[INFO] -----< edu.eci.cvds:Patterns >-----
[INFO] Building Patterns 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO] --- exec:3.5.0:java (default-cli) @ Patterns ---
Hello Esteban Aguilera
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.273 s
[INFO] Finished at: 2025-01-30T15:27:56-05:00
[INFO] -----
C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns>
```

```

C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns>cd ..
C:\Users\esteban.aguilera-c.LABINFO\Documents\lab>cd C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns
C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns>mkdir shapes
C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns>cd shapes
C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns\shapes>mkdir concrete
C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns\shapes>notepad Shape.java

C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns\shapes>ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns\shapes>dir
Volume in drive C has no label.
Volume Serial Number is 9437-6886

Directory of C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns\shapes

01/30/2025  03:33 PM  <DIR>          .
01/30/2025  03:33 PM  <DIR>          ..
01/30/2025  03:33 PM  <DIR>          concrete
01/30/2025  03:33 PM                0 Shape.java
               1 File(s)                0 bytes
               3 Dir(s)  852,480,577,536 bytes free

C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns\shapes>notepad RegularShapeType.java
C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns\shapes>cd concrete

C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns\shapes\concrete>notepad Hexagon.java
C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns\shapes\concrete>notepad Pentagon.java
C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns\shapes\concrete>notepad Quadrilateral.java
C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns\shapes\concrete>notepad Triangle.java

C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns\shapes\concrete>
C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns\shapes>notepad ShapeMain.java
C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns\shapes>notepad ShapeFactory.java
C:\Users\esteban.aguilera-c.LABINFO\Documents\lab\Patterns\src\main\java\edu\eci\cvds\patterns\shapes>

```