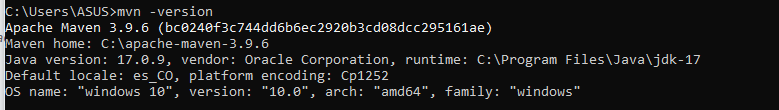
Laboratorio 2 CVDS

Sara Castillo

Nicolas Achuri

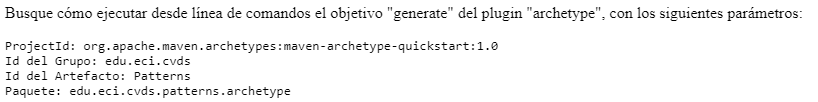
Tomas Suarez

Luego de instalar maven comprobamos su version con mvn -version





mvn archetype:generate



Lo hacemos con el comando mvn archetype:generate -DgroupId=edu.eci.cvds -DartifactId=Patterns -DarchetypeGroupId=org.apache.maven.archetypes -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.0 -DinteractiveMode=false -Dpackage=edu.eci.cvds.patterns.archetype

Este comando utiliza la opción -D para proporcionar valores específicos a las propiedades del arquetipo que se está utilizando. Aquí hay una breve explicación de cada parámetro:

-DgroupId: Especifica el identificador del grupo para el proyecto (edu.eci.cvds).

-DartifactId: Especifica el identificador del artefacto para el proyecto (Patterns).

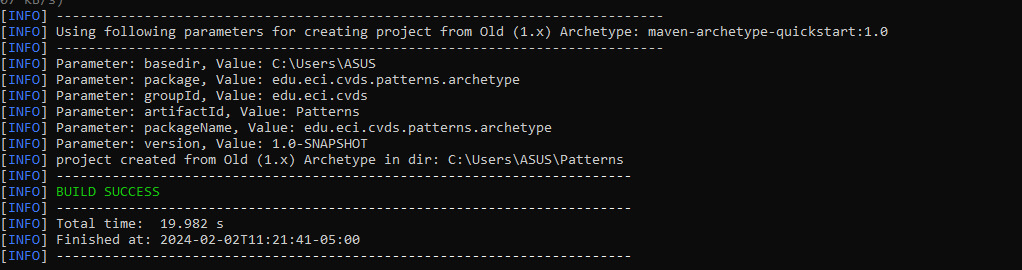
-DarchetypeGroupId: Especifica el identificador del grupo del arquetipo que se está utilizando (org.apache.maven.archetypes).

-DarchetypeArtifactId: Especifica el identificador del artefacto del arquetipo (maven-archetype-quickstart).

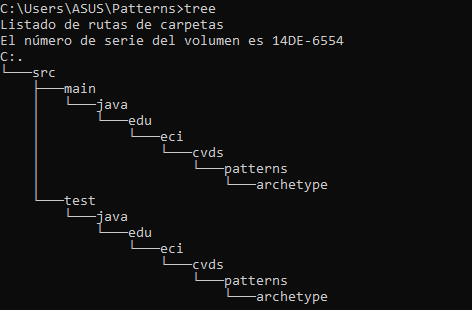
-DarchetypeVersion: Especifica la versión del arquetipo (1.0).

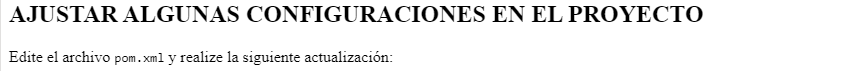
-DinteractiveMode=false: Deshabilita el modo interactivo para que Maven use los valores proporcionados sin pedir confirmación.

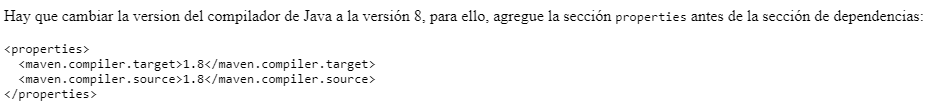
-Dpackage: Especifica el paquete para el proyecto (edu.eci.cvds.patterns.archetype).

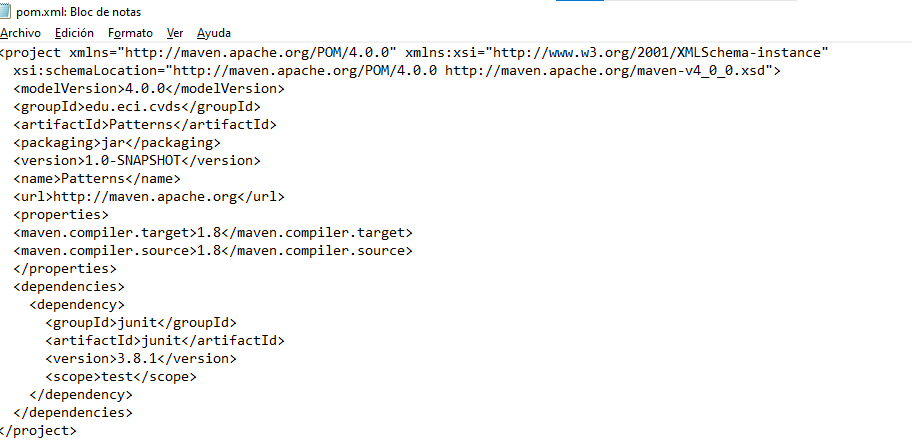


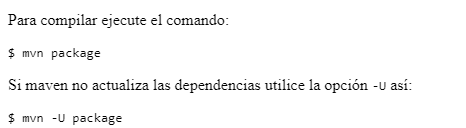


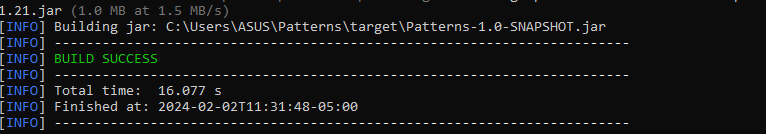














El parámetro "package" en el comando Maven (-Dpackage) se utiliza para especificar el nombre del paquete Java que se utilizará para organizar las clases dentro del proyecto generado. Este valor se utiliza para definir la estructura del paquete en el código fuente del proyecto Maven. Por ejemplo, si proporcionamos -Dpackage=edu.eci.cvds.patterns.archetype, las clases generadas estarán dentro del paquete edu.eci.cvds.patterns.archetype.

Aparte del parámetro "package", hay otros parámetros que puedes enviar al comando mvn archetype:generate para personalizar la creación del proyecto. Algunos de los parámetros más comunes incluyen:

-DgroupId: Especifica el identificador del grupo para el proyecto.

-DartifactId: Especifica el identificador del artefacto para el proyecto.

-Dversion: Especifica la versión del proyecto.

-DarchetypeGroupId: Especifica el identificador del grupo del arquetipo.

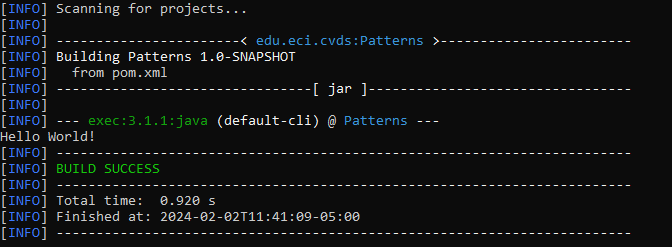
-DarchetypeArtifactId: Especifica el identificador del artefacto del arquetipo.

-DarchetypeVersion: Especifica la versión del arquetipo.

-DinteractiveMode=false: Deshabilita el modo interactivo para que Maven use los valores proporcionados sin pedir confirmación.

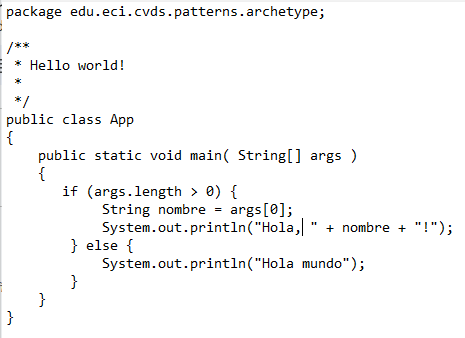


mvn exec:java -Dexec.mainClass="edu.eci.cvds.patterns.archetype.App"

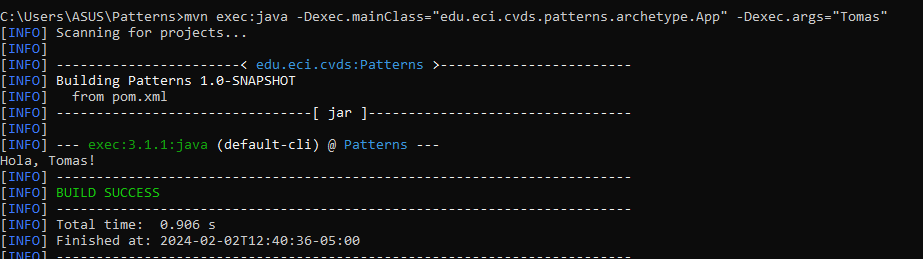




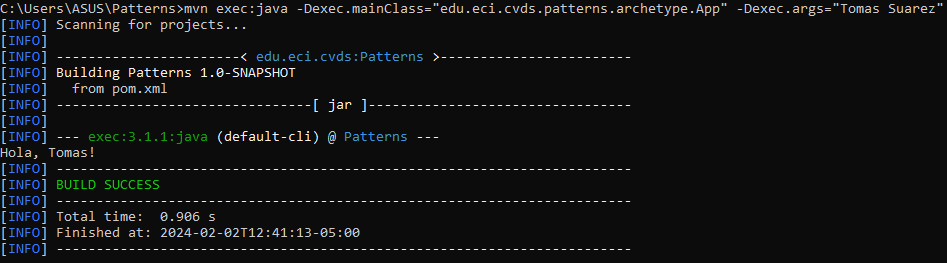
1. Ejecutamos mvn Package para compilar
2. Ejecutamos el comando anterior



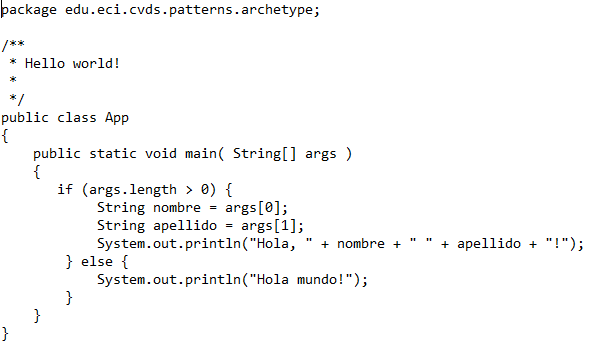


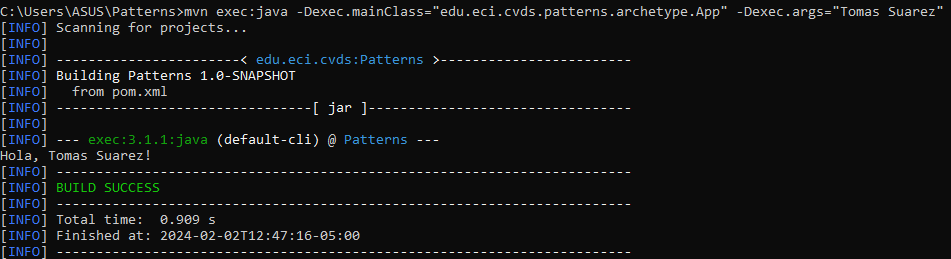




Aparece únicamente el nombre:  






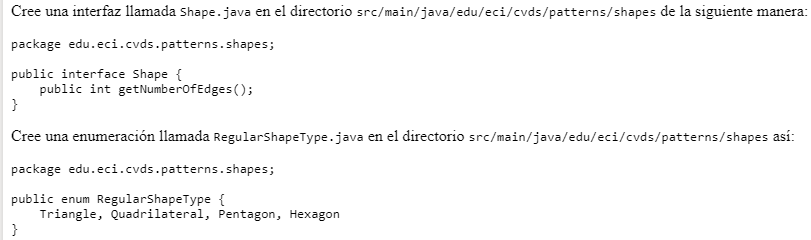


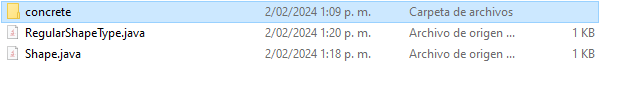
## **HACER EL ESQUELETO DE LA APLICACIÓN**

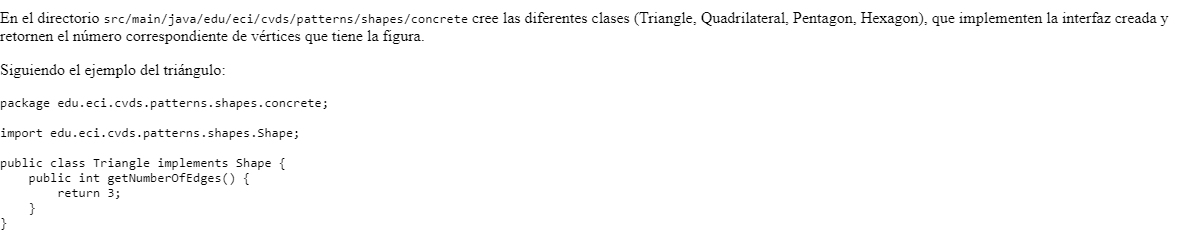


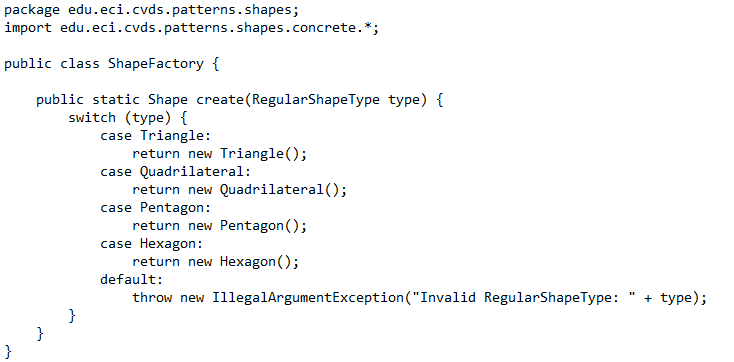




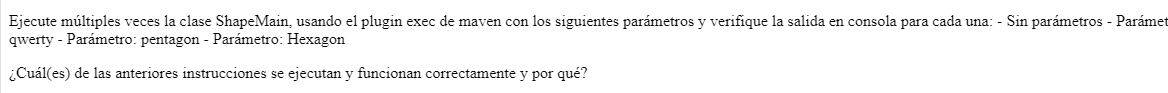


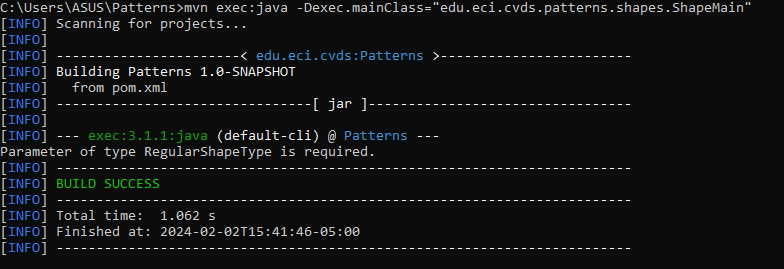


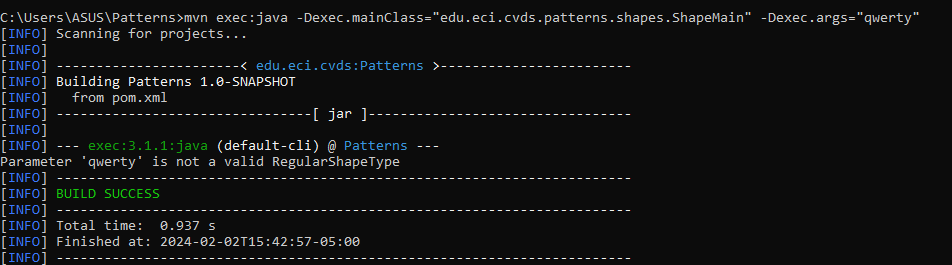




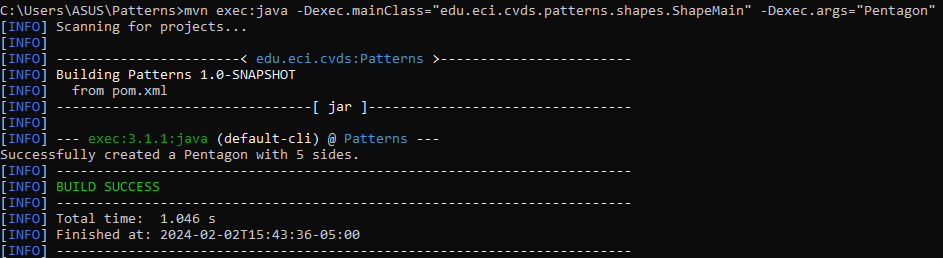
Usamos el patron de simple factory porque la lógica de creación de objetos (Shape) era relativamente directa y no requería la participación de subclases. La fábrica simple proporciona una implementación sencilla y directa para crear instancias de objetos basadas en un tipo dado (RegularShapeType).



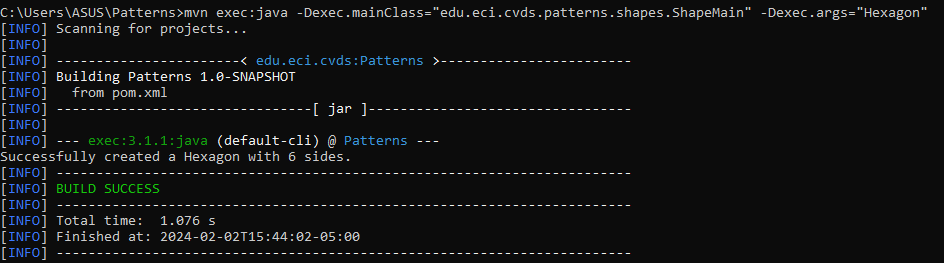
Sin argumentos:  


Parámetro qwerty:  


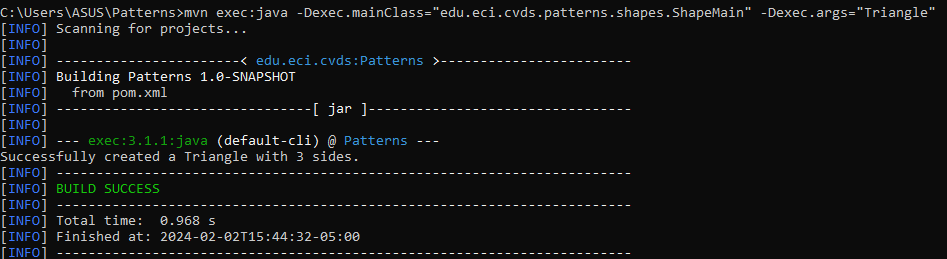
Parámetro Pentagon



Parámetro Hexagon



Parámetro Triangle



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

Las que tengan argumentos que hagan parte del enum funcionan ya que son válidos pero si pasamos un argumento vacio o que no pertenezca obtendremos la salida Parameter is not a valid RegularShapeType