**LABORATORIO 2 - PATTERNS - 2021-1**

Archivo que documenta y describe el proceso que se llevó a cabo en el laboratorio para poder desarrollar todas las actividades requeridas.

**CREAR UN PROYECTO CON MAVEN**

Configuramos el directorio en el que vamos a crear nuestro proyecto.

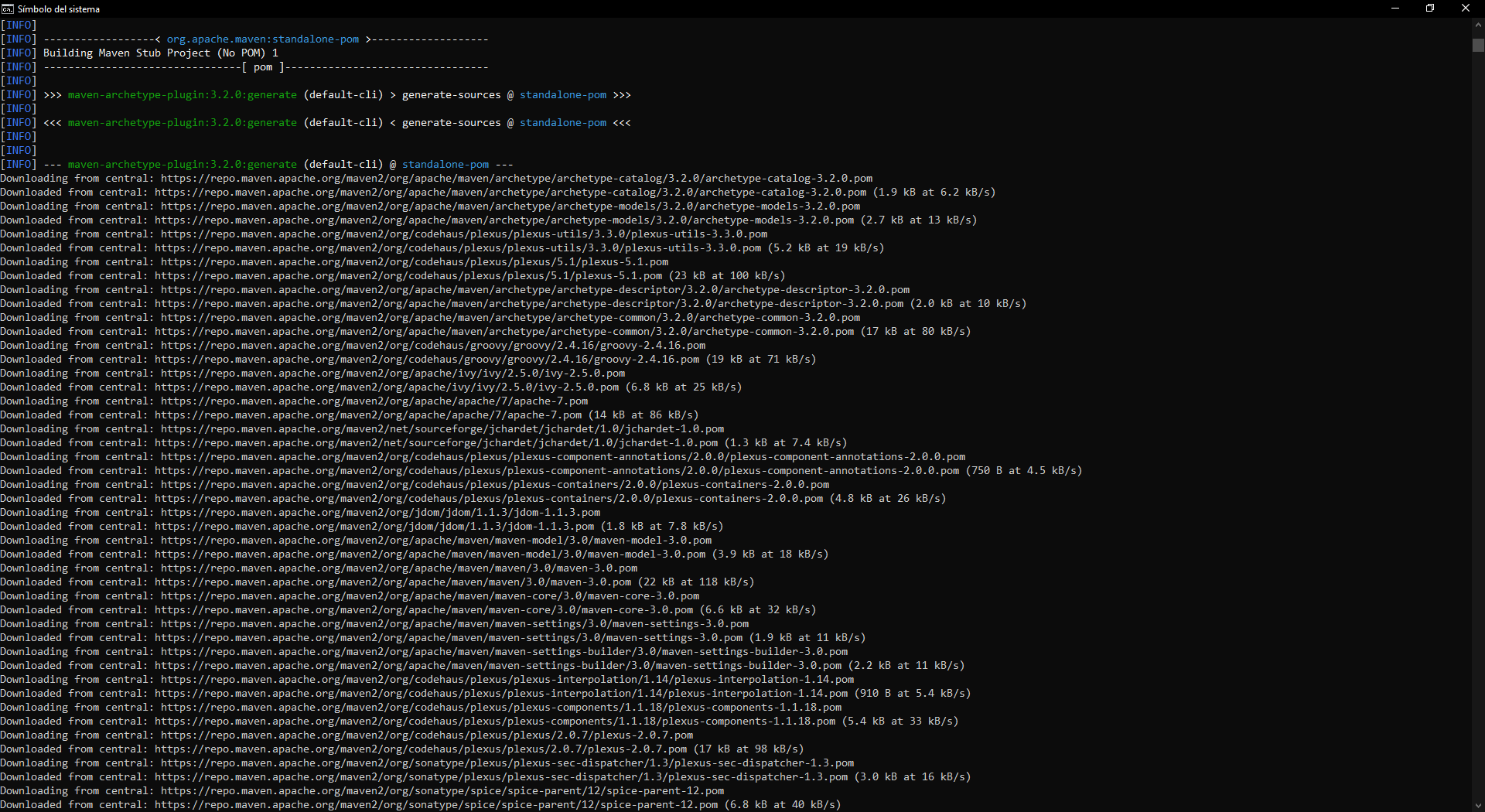
Luego, accedemos a la consola de comandos y nos situamos en el directorio previamente configurado e insertamos el siguiente comando:

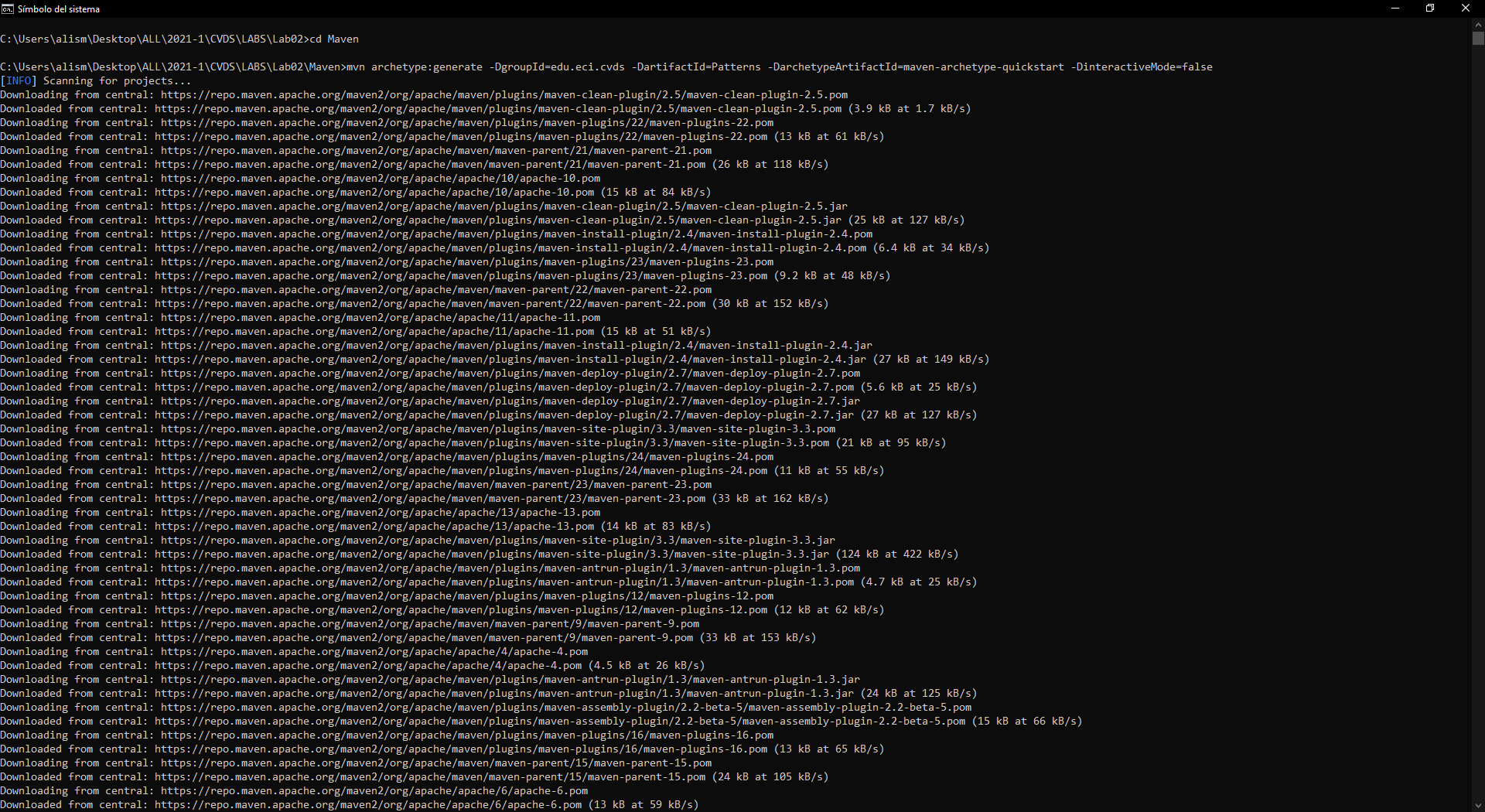
mvn archetype:generate -DgroupId=edu.eci.cvds -DartifactId=Patterns -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

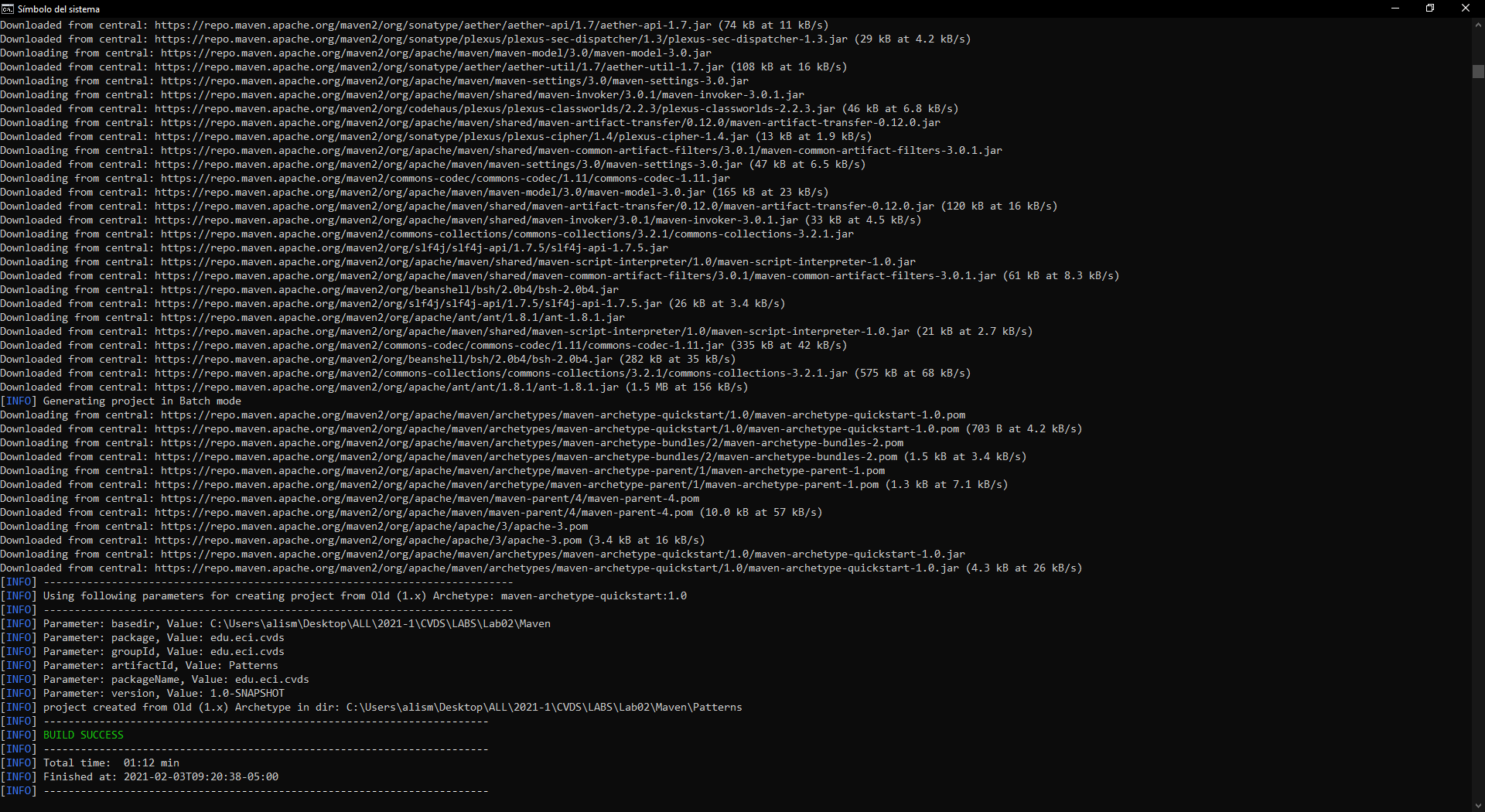
El comando viene dispuesto con ciertos valores que ya nos indicaron:

* Grupo: edu.eci.cvds
* Id del Artefacto: Patterns
* Paquete: edu.eci.cvds.patterns
* archetypeArtifactId: maven-archetype-quickstart

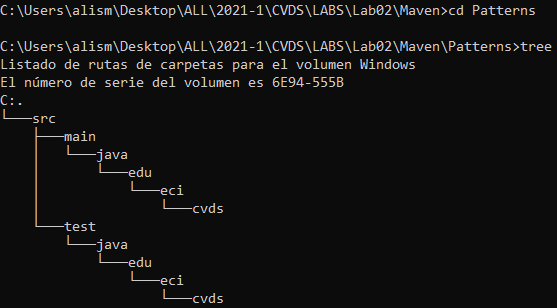
Al ejecutar el comando Maven empezará a descargar algunos recursos y a generar los directorios necesarios para armar la estructura de proyectos.







Ahora, cambiamos de directorio a Patterns, y ejecutamos el comando Tree, el cual nos mostrará la jerarquía de elementos en el proyecto.



**AJUSTAR ALGUNAS CONFIGURACIONES EN EL PROYECTO**

Modificaremos el archivo pom.xml que contiene las configuraciones del proyecto para indicar que vamos a ajustar la versión del compilador de Java a la 8.



Agregamos la sección properties justo antes de la sección dependencias.

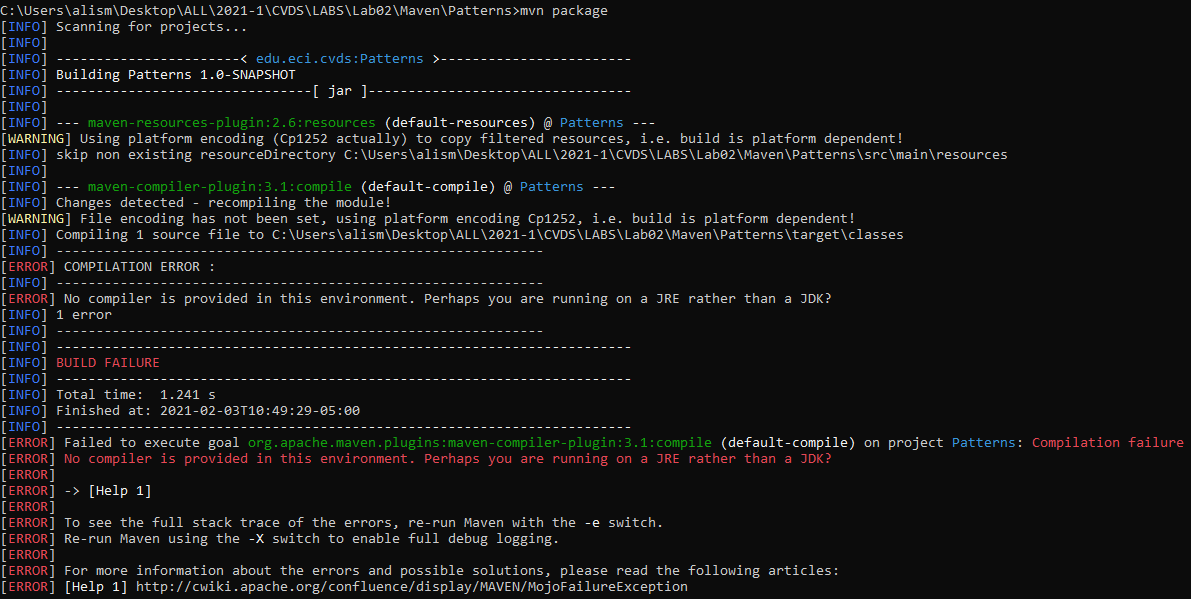
Después de haber realizado la modificación, es recomendable salir y volver a entrar a la línea de comandos, para que así se puedan ver reflejados los cambios y no nos valla a presentar inconvenientes más adelante.

**COMPILAR Y EJECUTAR**

Para poder empaquetar o generar el paquete de nuestro trabajo, ejecutamos el comando:

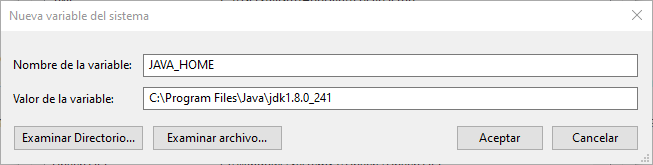
mvn package

Si nos sale un error como el siguiente

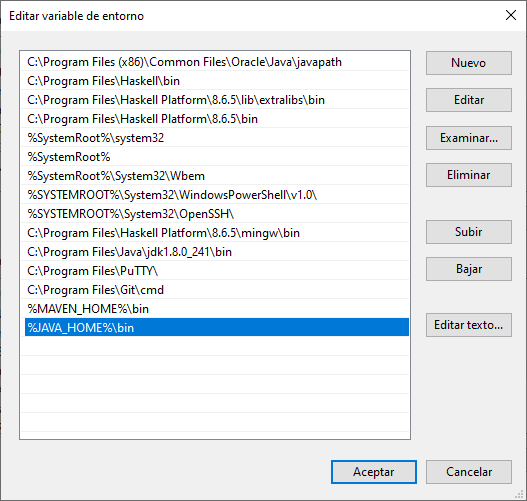


Tendremos que configurar la variable de entorno para java.

La creamos:



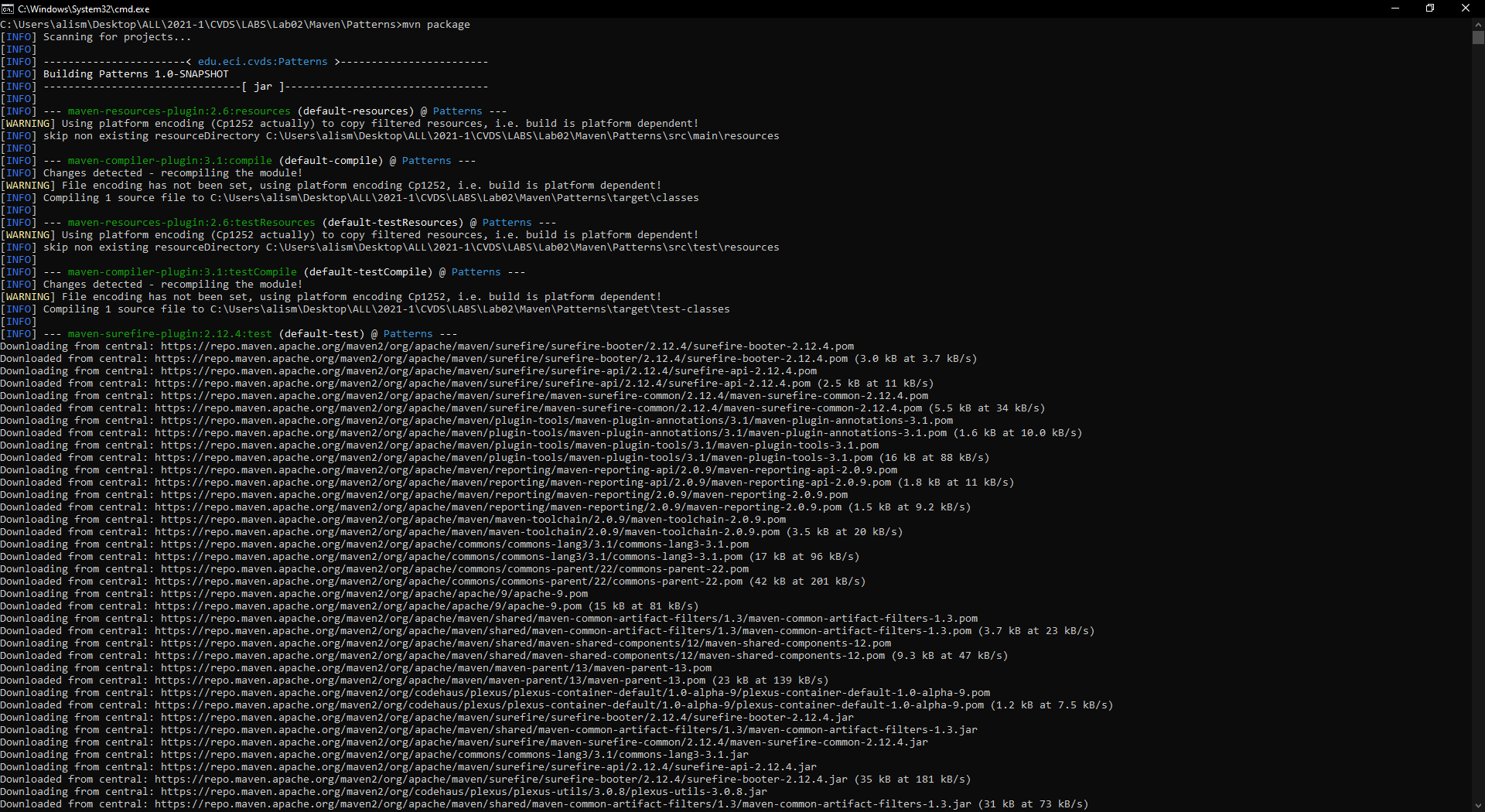
Y la agregamos al PATH (Del Sistema):

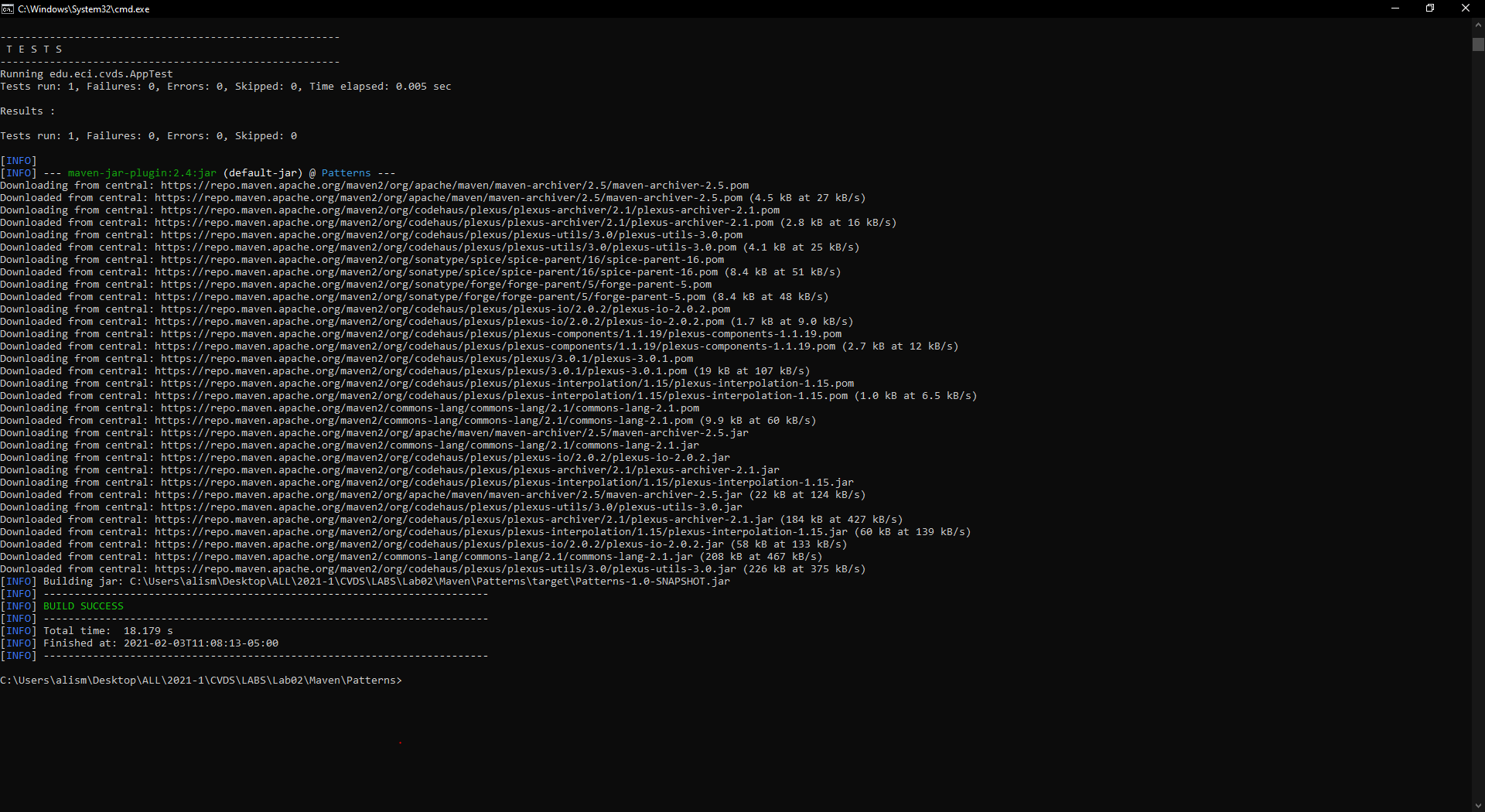


Aceptamos, y de nuevo volvemos a salir y entrar a la línea de comandos.

Ahora ya nos va a permitir empaquetar.

También va a descargar algunos recursos si es la primera vez que lo ejecutamos.





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

package permite realizar el empaquetamiento del proyecto.

Otros parámetros que también podemos ponerle a mvn son:

* mvn compile: Compila el proyecto.
* mvn test: Compila los test y los ejecuta.
* mvn package: Empaqueta el proyecto.
* mvn install: Guarda el proyecto en el repositorio.
* mvn clean: borra el directorio de salida (target).
* **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".**

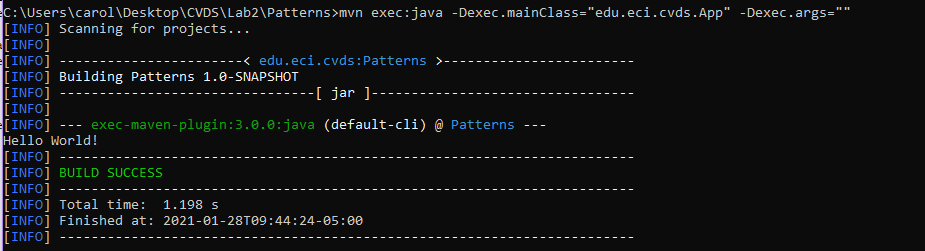
Para poder ejecutar una clase en Maven, usamos el siguiente comando:

mvn exec:java -Dexec.mainClass=”nombreArchivo” -Dexec.args=”argumento1 argumento2 …”

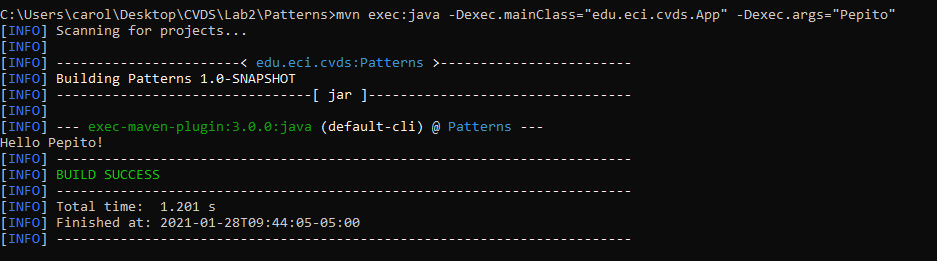
La última parte del comando se puede omitir si no hay argumentos.

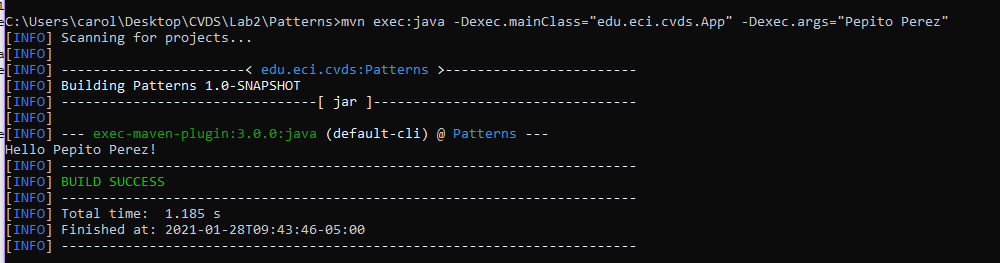
Para nuestro caso los comandos serían:

mvn exec:java -Dexec.mainClass=”edu.eci.cvds.App” -Dexec.args=””



Probamos el programa con los distintos parámetros y las modificaciones requeridas:

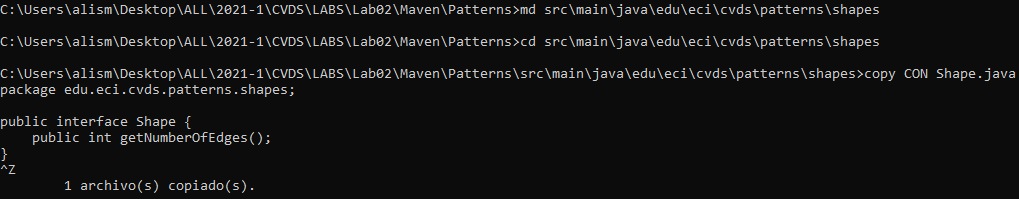




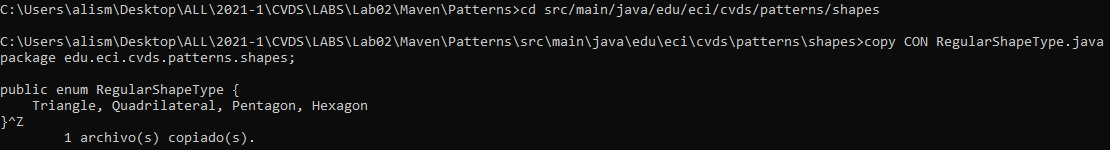
**HACER EL ESQUELETO DE LA APLICACION**

Ejecutamos los comandos en consola como se muestra a continuación par poder crear cada uno de los archivos en los directorios indicados:

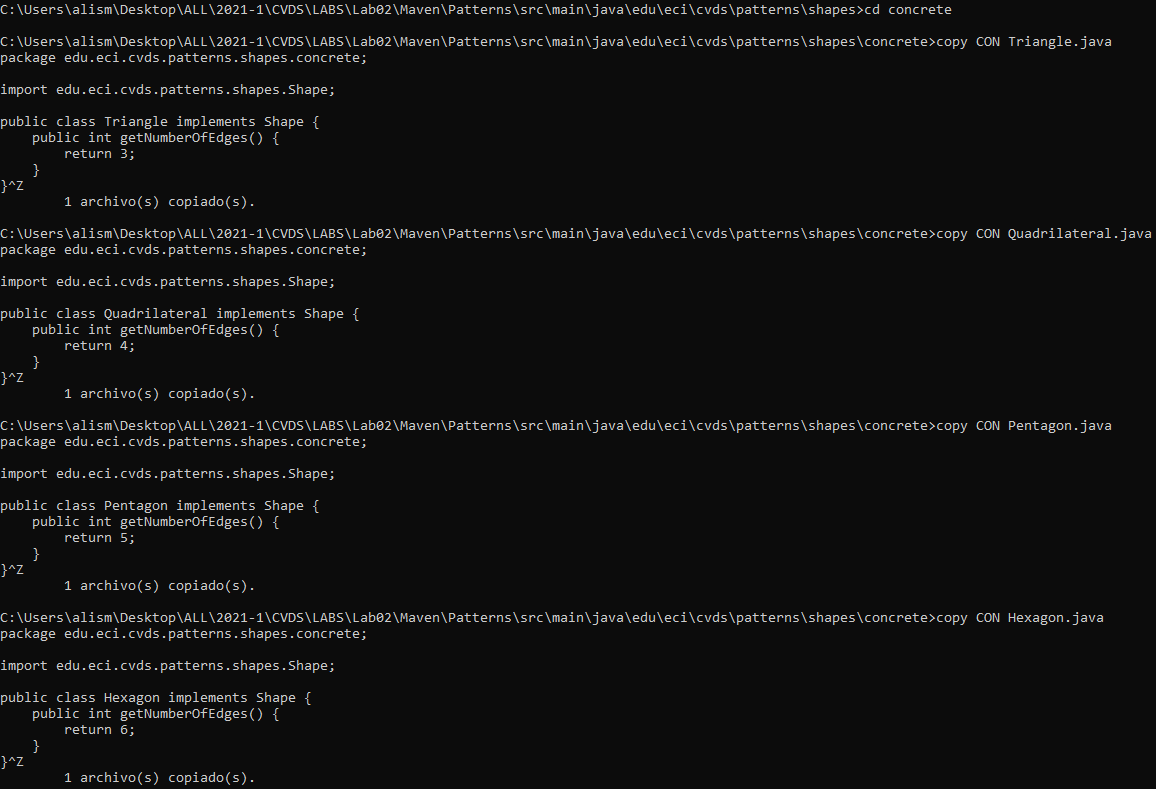
* Primera parte (Shape.java y paquetes nuevos)



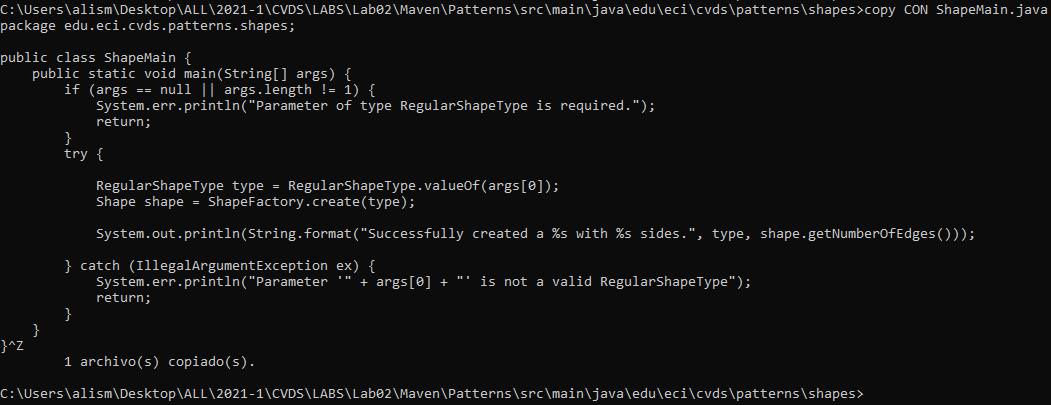
* Segunda parte (RegularShapeType.java)



* Tercera Parte (Triangle.java, Quadrilateral.java, Pentagon.java, Hexagon.java)



* Cuarta parte (ShapeMain.java)

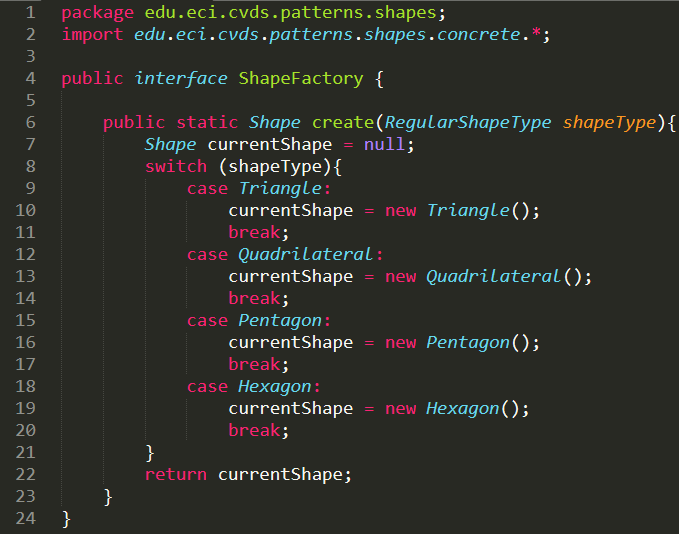


* Quinta parte (ShapeFactory.java)

Solo lo creamos pero lo dejamos vacío para poder editarlo más cómodamente en un editor de texto.

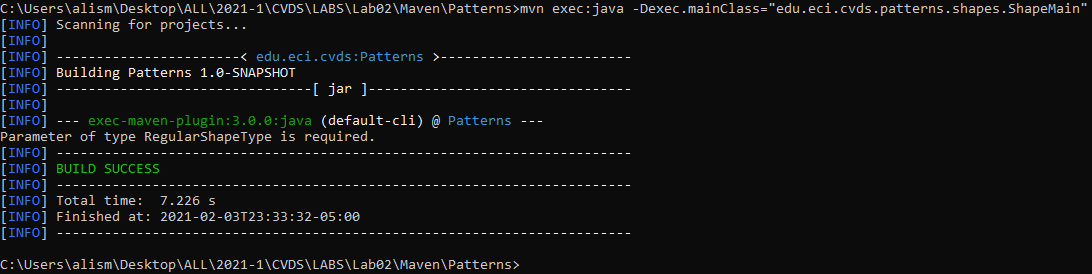


Al implementar **Factory Method**, la clase ShapeFactory queda de la siguiente manera:

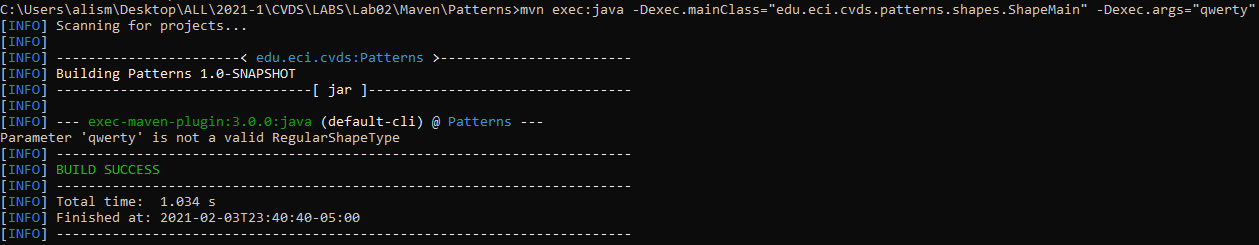


**Probando casos**

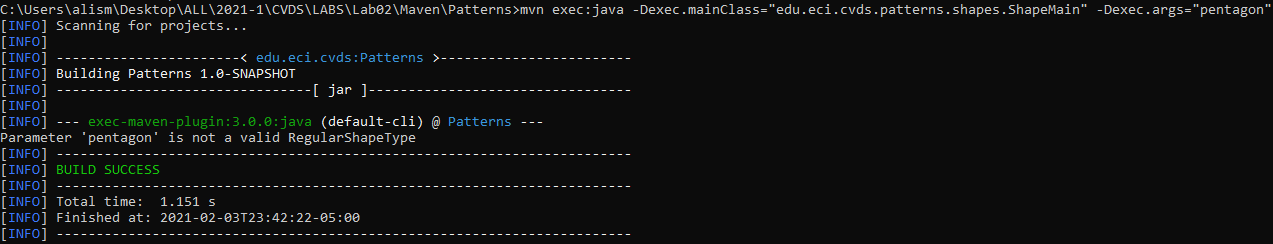
* Sin parámetros



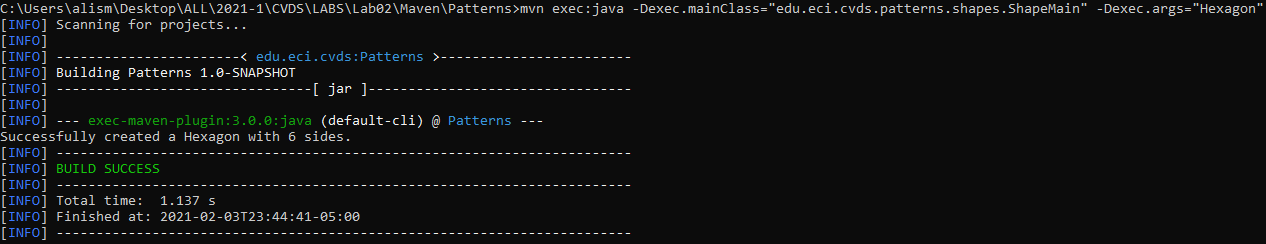
* Parámetro: qwerty



* Parámetro: pentagon



* Parámetro Hexagon



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

Cuando se ejecuta la instrucción sin parámetros, o con los parámetros qwerty o pentagon, la operación nos lanza una excepción.

Para el caso de cuando no hay argumentos, la excepción nos dice que debemos indicar el nombre de una figura. Cuando mandamos el parámetro qwerty la excepción se lanza porque este no es un caso contemplado en la clase enum. Y cuando el parámetro es pentagon la excepción se lanza porque aunque este caso es válido, está mal escrito, ya que debe iniciar con mayúscula para coincidir con algún caso del enum.

Finalmente, cuando se pasa el parámetro Hexagon la ejecución es correcta, ya que es un caso contemplado que coincide con uno del enum.