

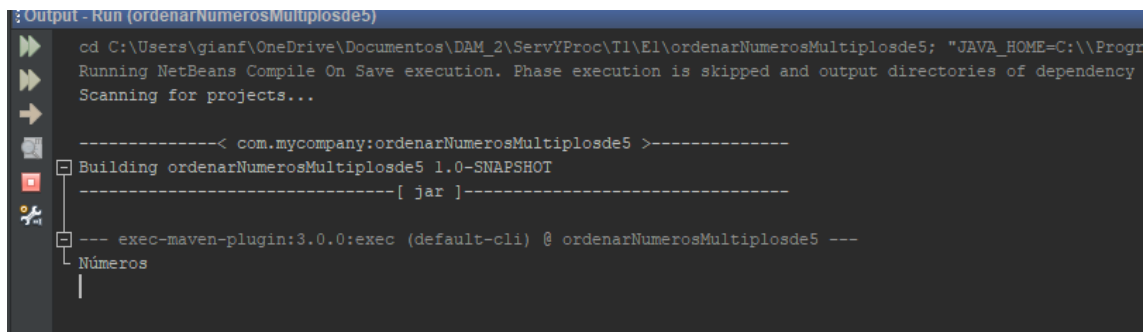
Pruebas PSP01

Índice

1. Prueba de aplicación 1 ordenación de números.....	3
2. Prueba generación de números aleatorios primos.....	3
3. Prueba de “tubería”	4

1. Prueba de aplicación 1 ordenación de números

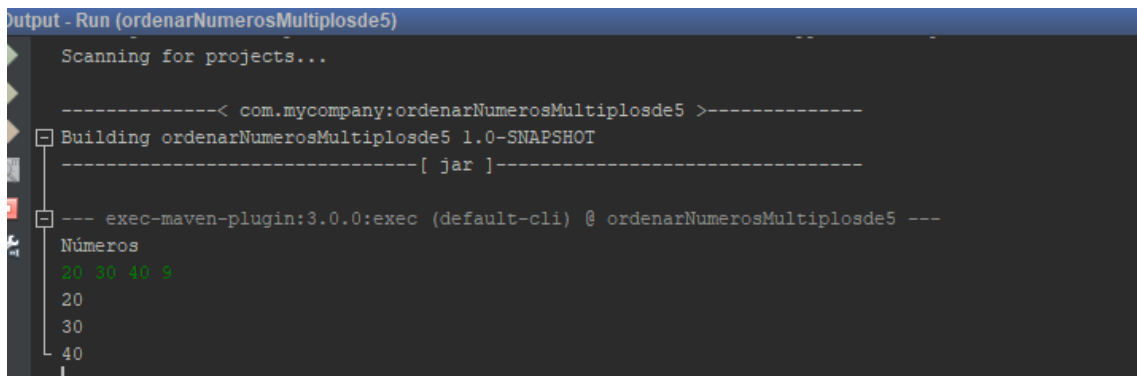
En esta prueba capto por consola una lista de números en una línea, el programa diferencia un número cada vez que haya un espacio:



```
Output - Run (ordenarNumerosMultiplosde5)
cd C:\Users\gianf\OneDrive\Documentos\DAM_2\ServYProc\T1\EI\ordenarNumerosMultiplosde5; "JAVA_HOME=C:\Progra
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency
Scanning for projects...

-----< com.mycompany:ordenarNumerosMultiplosde5 >-----
Building ordenarNumerosMultiplosde5 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ ordenarNumerosMultiplosde5 ---
Números
|
```



```
Output - Run (ordenarNumerosMultiplosde5)
Scanning for projects...

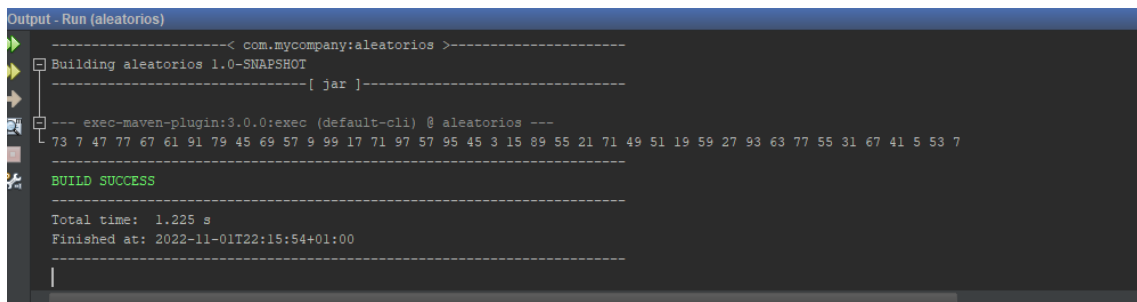
-----< com.mycompany:ordenarNumerosMultiplosde5 >-----
Building ordenarNumerosMultiplosde5 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ ordenarNumerosMultiplosde5 ---
Números
20 30 40 5
20
30
40
|
```

Al pulsar entrada el programa devuelve los números múltiplos de 5 ordenados de menor a mayor.

2. Prueba generación de números aleatorios primos

En esta prueba simplemente se ejecuta el programa que genera los 40 números aleatorios que se piden:



```
Output - Run (aleatorios)
-----< com.mycompany:aleatorios >-----
Building aleatorios 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ aleatorios ---
73 7 47 77 67 61 91 79 45 69 57 9 99 17 71 97 57 95 45 3 15 89 55 21 71 49 51 19 59 27 93 63 77 55 31 67 41 5 53 7

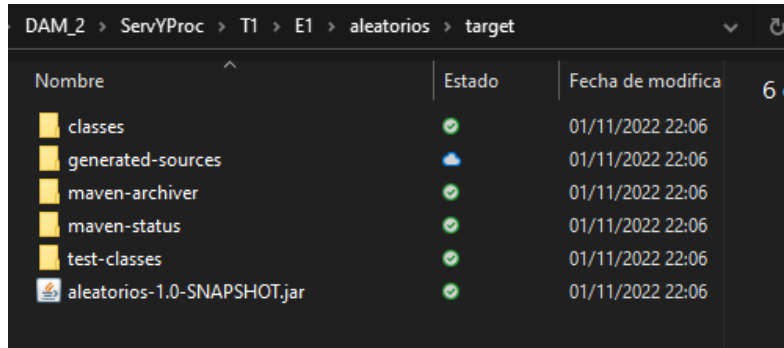
BUILD SUCCESS

Total time: 1.225 s
Finished at: 2022-11-01T22:15:54+01:00
|
```

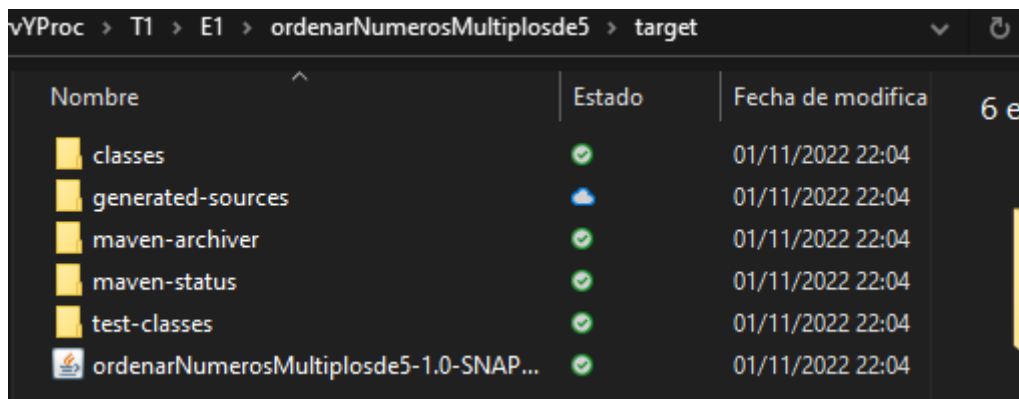
3. Prueba de “tubería”

En esta prueba se juntan las dos aplicaciones.

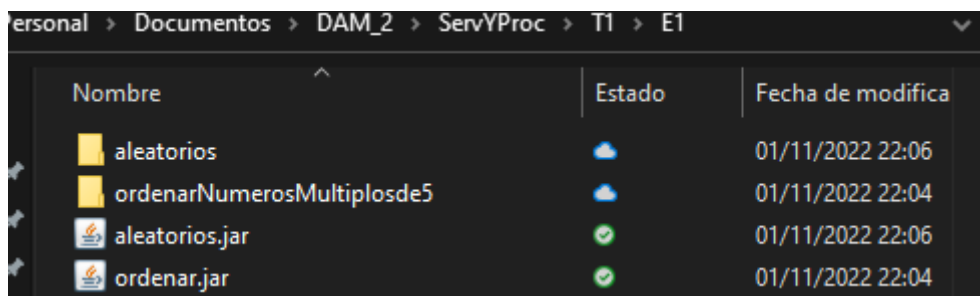
Primero generaremos los .jar los ubicaremos en la carpeta target de cada proyecto pero lo situaremos en la misma carpeta por comodidad:



Nombre	Estado	Fecha de modifica
classes	✓	01/11/2022 22:06
generated-sources	☁	01/11/2022 22:06
maven-archiver	✓	01/11/2022 22:06
maven-status	✓	01/11/2022 22:06
test-classes	✓	01/11/2022 22:06
aleatorios-1.0-SNAPSHOT.jar	✓	01/11/2022 22:06

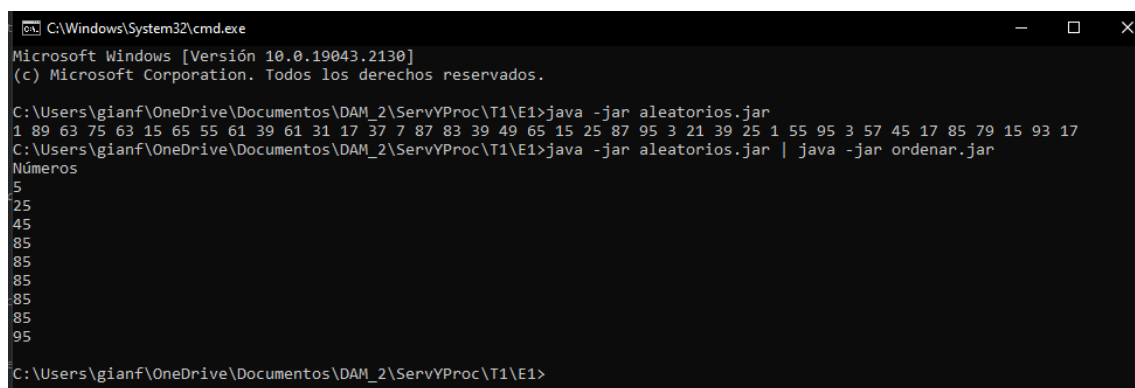


Nombre	Estado	Fecha de modifica
classes	✓	01/11/2022 22:04
generated-sources	☁	01/11/2022 22:04
maven-archiver	✓	01/11/2022 22:04
maven-status	✓	01/11/2022 22:04
test-classes	✓	01/11/2022 22:04
ordenarNumerosMultiplosde5-1.0-SNAP...	✓	01/11/2022 22:04



Nombre	Estado	Fecha de modifica
aleatorios	☁	01/11/2022 22:06
ordenarNumerosMultiplosde5	☁	01/11/2022 22:04
aleatorios.jar	✓	01/11/2022 22:06
ordenar.jar	✓	01/11/2022 22:04

Mediante consola captamos el resultado de aleatorio.jar que será tratado por el programa de ordenar.jar. Lo hacemos mediante el comando ***java -jar origen_datos | java -jar destino_datos***



```

C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19043.2130]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\gianf\OneDrive\Documentos\DAM_2\ServYProc\T1\E1>java -jar aleatorios.jar
1 89 63 75 63 15 65 55 61 39 61 31 17 37 7 87 83 39 49 65 15 25 87 95 3 21 39 25 1 55 95 3 57 45 17 85 79 15 93 17
C:\Users\gianf\OneDrive\Documentos\DAM_2\ServYProc\T1\E1>java -jar aleatorios.jar | java -jar ordenar.jar
Números
5
25
45
85
85
85
85
85
95

C:\Users\gianf\OneDrive\Documentos\DAM_2\ServYProc\T1\E1>

```

Podemos ver como se han ordenado los múltiplos de 5 de los primos aleatorios generados.