

The image shows a Visual Studio Code window with the following components:

- Explorer:** Shows the project structure. The file `execute.py` is selected under the `DISTRIBUTEDMATRIXMUL` folder. Other files include `.gitignore`, `DistributedMatrixMul.class`, `DistributedMatrixMul.java`, `Matrix.class`, `Matrix.java`, `Node.class`, `Node.java`, `NodeZero.class`, `NodeZero.java`, and `NodeZero$ConexionHandler.class`.
- Editor:** Displays the `execute.py` script. The script is a Python program that uses `multiprocessing.Pool` to execute four instances of the `java DistributedMatrixMul` command. It includes comments in Spanish explaining the purpose of the program and the author's name.
- Terminal:** Shows the command `javac .\DistributedMatrixMul.java` being executed in a PowerShell window. The output shows the command being run successfully.

```
1  #/*****
2  # *
3  # * Autor: Andrés Rodarte López
4  # * Programa para ejecutar más rapido los nodos
5  # * Del programa DistributedMatrixMul
6  # * Es muy lento
7  # *
8  # *****/
9  from multiprocessing import Pool
10 from multiprocessing.dummy import Pool as ThreadPool
11 from subprocess import Popen, PIPE, call
12
13
14 cmds = ["java DistributedMatrixMul 4", "java DistributedMatrixMul", "java DistributedMatrixMul", "java DistributedMatrixMul"]
15 def function_create_cmds(cmd):
16     proc = Popen(cmd, shell=True, stdout=PIPE, stderr=PIPE)
17     (output, error) = proc.communicate()
18     return output
19
20 # Make the Pool of workers
21 pool = ThreadPool(4)
22 results = pool.map(function_create_cmds, cmds)
23 #close the pool and wait for the work to finish
24 pool.close()
25 pool.join()
26 for each in results:
27     each = each.decode("utf-8")
28     if len(each) > 0:
29         print("out: {}".format(each))
```

PS A:\Andres\Repositorios\Practicas\Distribuidos\DistributedMatrixMul> javac .\DistributedMatrixMul.java
PS A:\Andres\Repositorios\Practicas\Distribuidos\DistributedMatrixMul>

Compilación del Programa. Y descripción del script en Python para ejecutar las 4 instancias del mismo.

```
1 1 #/*****
2 2 # *
3 3 # * Autor: Andrés Rodarte López
4 4 # * Programa para ejecutar más rapido los nodos

PS A:\Andres\Repositorios\Practicas\Distribuidos\DistributedMatrix\Mul> javac .\DistributedMatrix\Mul.java
PS A:\Andres\Repositorios\Practicas\Distribuidos\DistributedMatrix\Mul> py .\execute.py
out: Size of Matrix: 4 with : 4 nodes

Matrix A is
0.0 1.0 2.0 3.0
2.0 3.0 4.0 5.0
4.0 5.0 6.0 7.0
6.0 7.0 8.0 9.0
-- -- -- -- --

Matrix B is
0.0 2.0 4.0 6.0
-1.0 1.0 3.0 5.0
-2.0 0.0 2.0 4.0
-3.0 -1.0 1.0 3.0
-- -- -- -- --

Server is listening on port 5050
A new client is connected : Socket[addr=/127.0.0.1,port=51654,localport=5050]
Assigning new thread for this client
Starting node: 1
A new client is connected : Socket[addr=/127.0.0.1,port=51653,localport=5050]
Assigning new thread for this client
Starting node: 2
A new client is connected : Socket[addr=/127.0.0.1,port=51655,localport=5050]
Assigning new thread for this client
Starting node: 3

Matrix C is
28.0 22.0 16.0 10.0
52.0 38.0 24.0 10.0
76.0 54.0 32.0 10.0
100.0 70.0 40.0 10.0
-- -- -- -- --
592

out: 1

out: 2

out: 3
```

Resultado de la ejecución del script. Imprime en pantalla las respuestas de cada instancia en el orden en el que se ejecutaron. El output 1. Como última línea ejecuta el checksum. Definido como la suma de cada celda. La matriz C es la respuesta, solo se muestra si N es lo suficientemente pequeña.

Visual Studio Code interface showing the execution of a Python script named `execute.py` in a distributed matrix environment.

The Explorer sidebar shows the project structure:

- EXPLORER
 - OPEN EDITORS
 - Java Overview
 - `execute.py` (M)
 - DISTRIBUTEDMATRIXMUL
 - `.gitignore`
 - `~$mpilación del Programa.docx`
 - `Compilación del Programa.docx` (U)
 - `Compilación del Programa.pdf` (U)
 - `DistributedMatrixMul.class`
 - `DistributedMatrixMul.java`
 - `execute.py` (M)
 - `Matrix.class`
 - `Matrix.java`
 - `Node.class`
 - `Node.java` (1)
 - `NodeZero.class`
 - `NodeZero.java`
 - `NodeZero$ConexionHandler.class`

The main editor shows the code for `execute.py`:

```
1  #/*****
2  # *
3  # * Autor: Andrés Rodarte López
```

The Output window displays the execution results, including a table of values:

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL
76.0	54.0	32.0	10.0
100.0	70.0	40.0	10.0
-- --	-- --	-- --	-- --
592			

The Terminal window shows the command prompt where the script was executed:

```
PS A:\Andres\Repositorios\Practicas\Distribuidos\DistributedMatrix\Mul> py .\execute.py
out: Size of Matrix: 500 with : 4 nodes
Server is listening on port 5050
A new client is connected : Socket[addr=/127.0.0.1,port=51698,localport=5050]
Assigning new thread for this client
Starting node: 1
A new client is connected : Socket[addr=/127.0.0.1,port=51699,localport=5050]
Assigning new thread for this client
Starting node: 2
A new client is connected : Socket[addr=/127.0.0.1,port=51697,localport=5050]
Assigning new thread for this client
Starting node: 3
2147483647

out: 2

out: 3

out: 1

PS A:\Andres\Repositorios\Practicas\Distribuidos\DistributedMatrix\Mul>
```

Ejecución con matriz de 500*500