



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

*”Tarea 3. Multiplicación distribuida de
matrices”*

Alumno:

Lara Cázares Jaime Arturo

Materia:

Desarrollo de sistemas distribuidos

Grupo:

4CM3

Profesor: Pineda Guerrero Carlos

Índice general

1.	Capturas de pantallas.	3
1.1.	N=4	3
1.2.	N=1000	5

1. Capturas de pantallas.

1.1. N=4

Se comienza compilando el programa `MultiplicaMatrizDistribuidoEficiente.java`, la Figura 9 muestra este primer paso en el cual se tiene un $N=4$.

```
D:\Documentos\ESCOM\Distribuidos\Tarea3>javac MultiplicaMatrizDistribuido_Eficiente.java
D:\Documentos\ESCOM\Distribuidos\Tarea3>
```

Figura 1: Compilación del programa.

Se prosigue con la ejecución, pasando como argumento 0, ya que el programa interpreta este número para saber que algoritmo usar dado a que la división de la matriz A y B es diferente en, además de que siendo el 0 se establece como **Servidor** y espera conexiones de los clientes. En la Figura 10 se observa al servidor esperando conexiones. Dado a que N es igual a 4, el servidor muestra las matrices a multiplicar, observar la Figura 10.

```
D:\Documentos\ESCOM\Distribuidos\Tarea3>java MultiplicaMatrizDistribuido_Eficiente 0
Nodo 0
N=4
Matriz A en servidor
0      1      2      3
2      3      4      5
4      5      6      7
6      7      8      9
Matriz B en servidor
0      2      4      6
-1     1      3      5
-2     0      2      4
-3    -1     1      3
```

Figura 2: Programa PI como **servidor** (nodo 0) en espera de conexiones de clientes para multiplicar matrices de 4×4 .

Se ejecutan los clientes, que es el mismo programa pero llevando a cabo distintos algoritmos dependiendo del entero pasado como argumento, se comienza iniciando conexión con el cliente y enviándole las matrices a multiplicar. El cliente multiplica las matrices recibidas y las envías de vuelta para que se asignen al segmento de la matriz C correspondiente. La Figura 3, 4, 5 y 6 muestran la ejecución de los nodos 1, 2, 3 y 4, respectivamente, estas ejecuciones no muestran ninguna salida a consola.

```
D:\Documentos\ESCOM\Distribuidos\Tarea3>java MultiplicaMatrizDistribuido_Eficiente 1
D:\Documentos\ESCOM\Distribuidos\Tarea3>
```

Figura 3: Programa como **cliente** (nodo 1).

```
D:\Documentos\ESCOM\Distribuidos\Tarea3>java MultiplicaMatrizDistribuido_Eficiente 2
D:\Documentos\ESCOM\Distribuidos\Tarea3>
```

Figura 4: Programa como **cliente** (nodo 2).

```
D:\Documentos\ESCOM\Distribuidos\Tarea3>java MultiplicaMatrizDistribuido_Eficiente 3
D:\Documentos\ESCOM\Distribuidos\Tarea3>
```

Figura 5: Programa como **cliente** (nodo 3).

```
D:\Documentos\ESCOM\Distribuidos\Tarea3>java MultiplicaMatrizDistribuido_Eficiente 4
D:\Documentos\ESCOM\Distribuidos\Tarea3>
```

Figura 6: Programa como **cliente** (nodo 4).

Una vez que los 4 nodos clientes terminan su ejecución, el nodo 0 que funge como **servidor** imprime en la consola la matriz C resultante de la multiplicación si y solo si N es igual a 4, calcula el valor de checksum, lo despliega y termina su ejecución. En la Figura 15 se aprecia el despliegue de la matriz C y el resultado del checksum.

```

D:\Documentos\ESCOM\Distribuidos\Tarea3>java MultiplicaMatrizDistribuido_Eficiente 0
Nodo 0
N=4
Matriz A en servidor
0      1      2      3
2      3      4      5
4      5      6      7
6      7      8      9
Matriz B en servidor
0      2      4      6
-1     1      3      5
-2     0      2      4
-3     -1     1      3
Matriz C:
28     22     16     10
52     38     24     10
76     54     32     10
100    70     40     10
Checksum=592

```

Figura 7: Programa como **servidor** (nodo 0) desplegando el resultado del calculo distribuido de la multiplicación de la matriz A y B, además del calculo de su checksum.

Por último en la Figura 8 se observa la captura de todas las terminales ejecutando el calculo distribuido de la multiplicación de matices.

```

PS D:\documentos\ESCOM\Distribuidos\Tarea3> java MultiplicaMatrizDistribuido_Eficiente 0
Nodo 0
N=4
Matriz A en servidor
0      1      2      3
2      3      4      5
4      5      6      7
6      7      8      9
Matriz B en servidor
0      2      4      6
-1     1      3      5
-2     0      2      4
-3     -1     1      3
Matriz C:
28     22     16     10
52     38     24     10
76     54     32     10
100    70     40     10
Checksum=592
PS D:\documentos\ESCOM\Distribuidos\Tarea3>
PS D:\documentos\ESCOM\Distribuidos\Tarea3> java MultiplicaMatrizDistribuido_Eficiente 1
PS D:\documentos\ESCOM\Distribuidos\Tarea3> []
PS D:\documentos\ESCOM\Distribuidos\Tarea3> java MultiplicaMatrizDistribuido_Eficiente 2
PS D:\documentos\ESCOM\Distribuidos\Tarea3> []
PS D:\documentos\ESCOM\Distribuidos\Tarea3> java MultiplicaMatrizDistribuido_Eficiente 3
PS D:\documentos\ESCOM\Distribuidos\Tarea3> []
PS D:\documentos\ESCOM\Distribuidos\Tarea3> java MultiplicaMatrizDistribuido_Eficiente 4
PS D:\documentos\ESCOM\Distribuidos\Tarea3> []

```

Figura 8: Captura de todas las terminales de la ejecución del programa con $N=4$, corrido en la terminal de Visual Studio Code.

1.2. $N=1000$

Se comienza compilando el programa MultiplicaMatrizDistribuidoEficiente.java, la Figura 9 muestra este primer paso en el cual se tiene un $N=1000$.

```
D:\Documentos\ESCOM\Distribuidos\Tarea3>javac MultiplicaMatrizDistribuido_Eficiente.java
D:\Documentos\ESCOM\Distribuidos\Tarea3>
```

Figura 9: Compilación del programa.

Se prosigue con la ejecución, pasando como argumento 0, ya que el programa interpreta este número para saber que algoritmo usar dado a que la división de la matriz A y B es diferente en, además de que siendo el 0 se establece como **Servidor** y espera conexiones de los clientes. En la Figura 10 se observa al servidor esperando conexiones. Dado a que N es igual a 1000, el servidor NO muestra las matrices a multiplicar, observar la Figura 10.

```
D:\Documentos\ESCOM\Distribuidos\Tarea3>java MultiplicaMatrizDistribuido_Eficiente 0
Nodo 0
N=1000
```

Figura 10: Programa PI como **servidor** (nodo 0) en espera de conexiones de clientes para multiplicar matrices de 1000x1000.

Se ejecutan los clientes, que es el mismo programa pero llevando a cabo distintos algoritmos dependiendo del entero pasado como argumento, se comienza iniciando conexión con el cliente y enviándole las matrices a multiplicar. El cliente multiplica las matrices recibidas y las envías de vuelta para que se asignen al segmento de la matriz C correspondiente. La Figura 11, 12, 13 y 14 muestran la ejecución de los nodos 1, 2, 3 y 4, respectivamente, estas ejecuciones no muestran ninguna salida a consola.

```
D:\Documentos\ESCOM\Distribuidos\Tarea3>java MultiplicaMatrizDistribuido_Eficiente 1
D:\Documentos\ESCOM\Distribuidos\Tarea3>
```

Figura 11: Programa como **cliente** (nodo 1).

```
D:\Documentos\ESCOM\Distribuidos\Tarea3>java MultiplicaMatrizDistribuido_Eficiente 2
D:\Documentos\ESCOM\Distribuidos\Tarea3>
```

Figura 12: Programa como **cliente** (nodo 2).

```
D:\Documentos\ESCOM\Distribuidos\Tarea3>java MultiplicaMatrizDistribuido_Eficiente 3
D:\Documentos\ESCOM\Distribuidos\Tarea3>
```

Figura 13: Programa como **cliente** (nodo 3).

```
D:\Documentos\ESCOM\Distribuidos\Tarea3>java MultiplicaMatrizDistribuido_Eficiente 4
D:\Documentos\ESCOM\Distribuidos\Tarea3>
```

Figura 14: Programa como **cliente** (nodo 4).

Una vez que los 4 nodos clientes terminan su ejecución, el nodo 0 que funge como **servidor** imprime en la consola la matriz *C* resultante de la multiplicación si y solo si *N* es igual a 4, calcula el valor de checksum, lo despliega y termina su ejecución. En la Figura 15 se aprecia el despliegue de la matriz *C* y el resultado del checksum.

```
D:\Documentos\ESCOM\Distribuidos\Tarea3>java MultiplicaMatrizDistribuido_Eficiente 0
Nodo 0
N=1000
Checksum=791635400632448
```

Figura 15: Programa como **servidor** (nodo 0) desplegando el resultado del calculo de su checksum.

Por último en la Figura 16 se observa la captura de todas las terminales ejecutando el calculo distribuido de la multiplicación de matices.

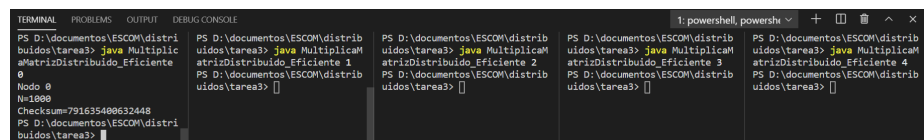


Figura 16: Captura de todas las terminales de la ejecución del programa con *N*=1000, corrido en la terminal de Visual Studio Code.