

**INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE COMPUTO**



**TAREA 3 MULTIPLICACIÓN DE MATRICES
DISTRIBUIDAS UTILIZANDO PASO DE
MENSAJES**

**MATERIA: DISEÑO DE SISTEMAS
DISTRIBUIDOS**

GRUPO: 4CV13

ALUMNO: MORA GUZMÁN JOSE ANTONIO

**FECHA ENTREGA: VIERNES 17 SEPTIEMBRE
2021**

INSTRUCCIONES DE LA TAREA

En esta tarea cada alumno deberá desarrollar **un solo programa** en Java, el cual calculará el producto de dos matrices cuadradas en forma distribuida sobre cinco nodos.

Sean A, B y C matrices cuadradas con elementos de tipo long, N renglones y N columnas, N par y $C=A \times B$.

Se deberá ejecutar dos casos:

1. N=10, desplegar las matrices A, B y C y el checksum de la matriz C.
2. N=1500, desplegar el checksum de la matriz C.

El checksum de la matriz C se calculará como la suma de todos elementos de la matriz. Para calcular la sumatoria se deberá declarar una variable "checksum" de tipo long.

$$\text{checksum} = \sum C[i][j], i=0, \dots, N-1, j=0, \dots, N-1.$$

Se deberá inicializar las matrices de la siguiente manera:

$$A[i][j] = 3*i+j$$

$$B[i][j] = 3*i-j$$

Donde $A[i][j]$ y $B[i][j]$ son los elementos A_{ij} y B_{ij} respectivamente.

El programa deberá ser ejecutado en cinco **máquinas virtuales con Ubuntu** (1 CPU, 1GB de RAM y disco HDD estándar) en cada máquina virtual se pasará como parámetro al programa el número de nodo, a saber: 0, 1, 2, 3 y 4.

El nombre de cada máquina virtual **deberá** ser una letra "M", el número de boleta del alumno, un guion y el número de nodo, por ejemplo, si el número de boleta del alumno es 12345678, entonces el nodo 0 deberá llamarse: M12345678-0, el nodo 1 deberá llamarse M12345678-1, y así sucesivamente. **No se admitirá la tarea** si los nodos no se nombran como se indicó anteriormente.

Recuerden que deben eliminar las máquinas virtuales cuando no las usen, con la finalidad de ahorrar el saldo de sus cuentas de Azure.

¿Cómo realizar la multiplicación de matrices en forma distribuida?

Suponga que divide la matriz A en las matrices A1 y A2. El tamaño de las matrices A1 y A2 es N/2 renglones y N columnas.

La matriz B se divide en las matrices B1 y B2. El tamaño de matrices B1 y B2 es N renglones y N/2 columnas.

Entonces la matriz $C=A \times B$ se compone de las matrices $C1$, $C2$, $C3$ y $C4$, tal como se muestra en la siguiente figura:



Donde:

$$C1 = A1 \times B1$$

$$C2 = A1 \times B2$$

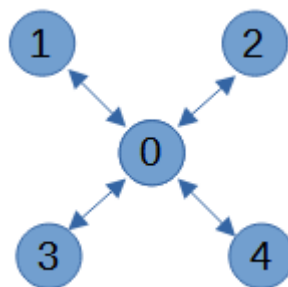
$$C3 = A2 \times B1$$

$$C4 = A2 \times B2$$

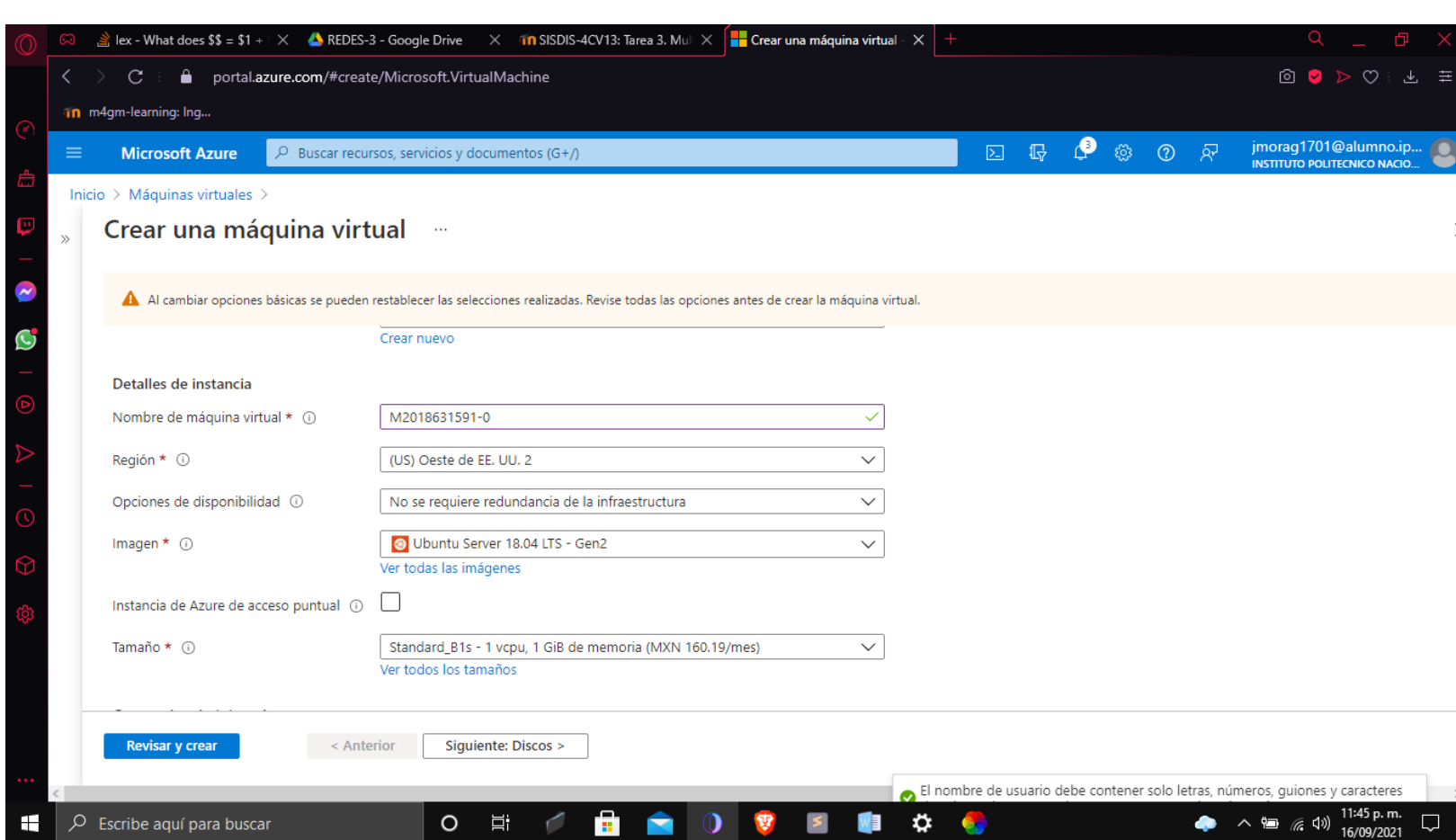
Debido a que las matrices se guardan en memoria por renglones, es más eficiente transponer la matriz B y dividirla de la siguiente manera:



Ahora supongamos que tenemos cinco nodos identificados con los números 0, 1, 2, 3 y 4.



CAPTURAS DE PANTALLA



En la captura anterior se muestra el nombre de la maquina virtual del nodo 0, la región la imagen y el tamaño (Este tamaño se modificó después por razones que explicare más adelante)

portal.azure.com/#create/Microsoft.VirtualMachine

m4gm-learning: Ing...

Microsoft Azure

Buscar recursos, servicios y documentos (G+)

Inicio > Máquinas virtuales >

Crear una máquina virtual

Al cambiar opciones básicas se pueden restablecer las selecciones realizadas. Revise todas las opciones antes de crear la máquina virtual.

Cuenta de administrador

Tipo de autenticación

☐ Clave pública SSH

☒ Contraseña

Nombre de usuario *

M2018631591-q

Contraseña *

Confirmar contraseña *

Reglas de puerto de entrada

Seleccione los puertos de red de máquina virtual que son accesibles desde la red Internet pública. Puede especificar acceso de red más limitado o granular en la pestaña Red.

Puertos de entrada públicos *

☐ Ninguno

☒ Permitir los puertos seleccionados

Revisar y crear

< Anterior

Siguiente: Discos >

Escribe aquí para buscar

11:45 p. m. 16/09/2021

En la captura anterior seleccionamos autenticador por contraseña y se colocó el nombre de usuario, la contraseña y se permitieron los puertos de entrada públicos seleccionados en este caso el SSH

A continuación se muestra en el apartado de discos que se seleccionó el disco HDD estándar

portal.azure.com/#create/Microsoft.VirtualMachine

m4gm-learning: Ing...

Microsoft Azure

Buscar recursos, servicios y documentos (G+)

Inicio > Máquinas virtuales >

Crear una máquina virtual

Datos básicos

Discos

Redes

Administración

Opciones avanzadas

Etiquetas

Revisar y crear

Las máquinas virtuales de Azure tienen un disco de sistema operativo y un disco temporal para el almacenamiento a corto plazo. Puede asociar discos de datos adicionales. El tamaño de la máquina virtual determina el tipo de almacenamiento que puede usar y la cantidad de datos que permiten los discos. [Más información](#)

Opciones de disco

Tipo de disco del sistema operativo *

HDD estándar (almacenamiento con redundancia local)

Elija discos SSD Premium para reducir la latencia, IOPS y el ancho de banda más altos, y expansión de disco. Las máquinas virtuales de instancia única con discos SSD Premium cumplen el SLA de conectividad de 99,9 %. [Más información](#)

Tipo de cifrado *

(Predeterminado) Cifrado en reposo con una clave administrada por la pl...

Habilitar compatibilidad con Ultra Disks

El disco Ultra se admite en las zonas de disponibilidad 1,2,3 para el tamaño de VM seleccionado (Standard_B1s).

Discos de datos

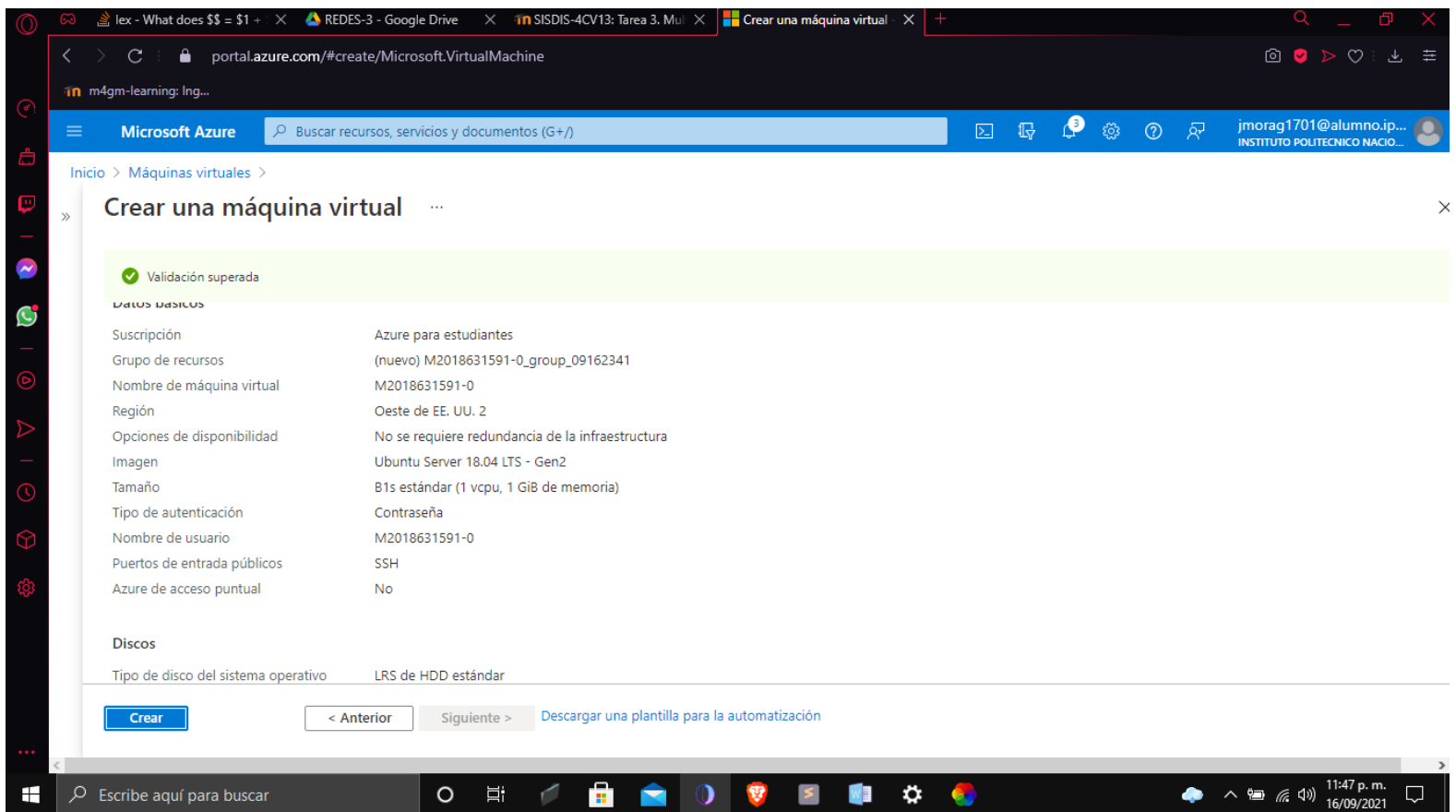
Revisar y crear

< Anterior

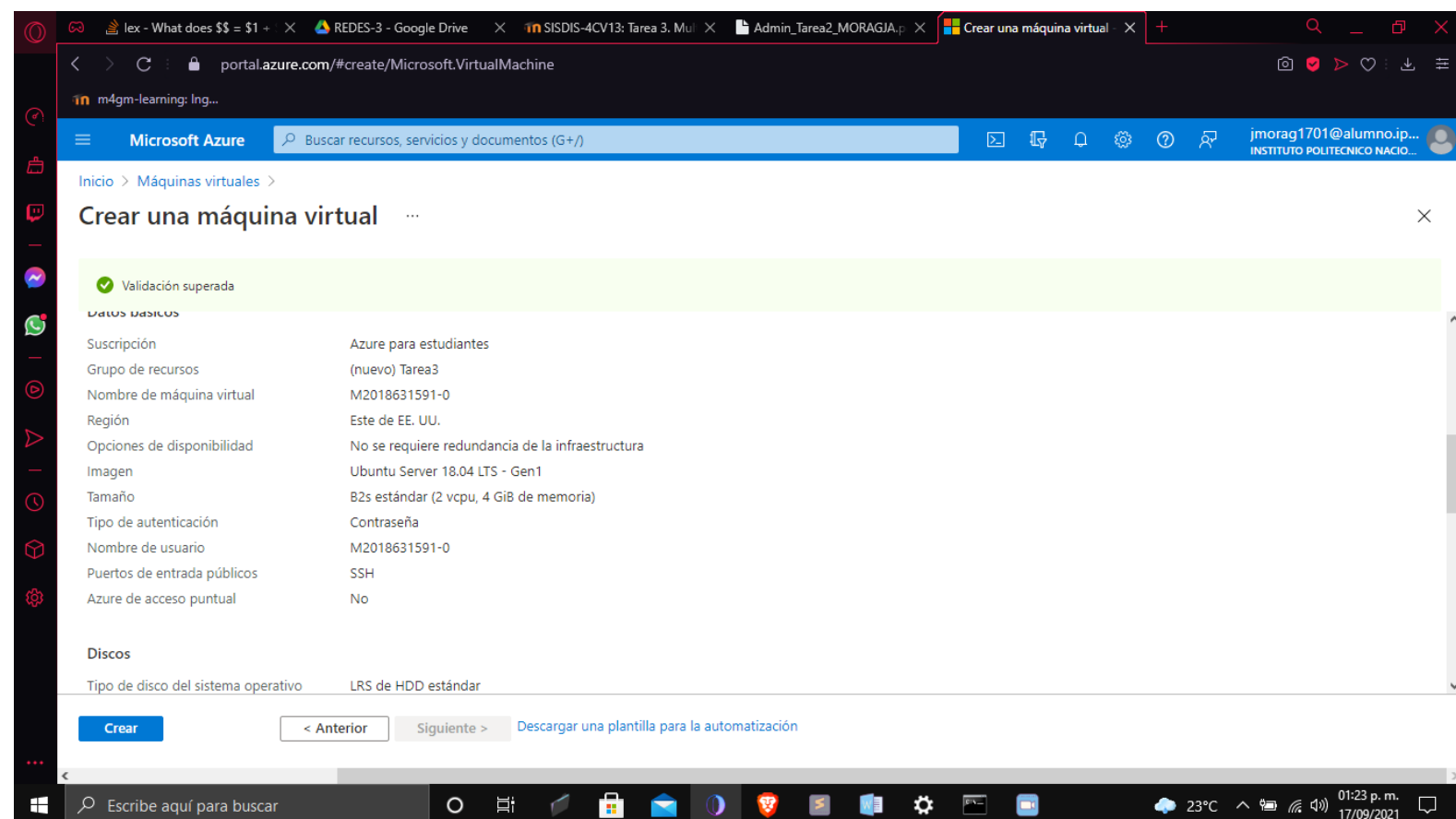
Siguiente: Redes >

Escribe aquí para buscar

11:46 p. m. 16/09/2021



Finalmente le dimos clic a revisar y crear donde nos arrojó la pantalla anterior. Hasta la parte anterior todo iba bien, pero al ejecutar el programa en esa maquina virtual con esas especificaciones, el programa se quedaba esperando y no respondía por lo que buscando soluciones encontré que era debido a la memoria así que cambié la memoria a 2vcpu y 4gb de RAM solamente para la maquina que hace de servidor en este caso el Nodo 0 (M2018631591-0), solucionando así el problema de que se quedaba esperando el programa.



Microsoft Azure

Inicio > CreateVm-Canonical.UbuntuServer-18_04-Its-gen2-20210916233501 > M2018631591-4

M2018631591-4 | Redes

Máquina virtual

Buscar (Ctrl+/)

Adjuntar interfaz de red Desasociar interfaz de red

Configuración de IP [ⓘ]

ipconfig1 (Principal)

Interfaz de red: m2018631591-4449 Reglas de seguridad vigentes Solucionar problemas de conexión de VM Topología

Red virtual/subred: M2018631591-4_group_09162347-vnet/default IP pública de NIC: M2018631591-4-ip IP privada de NIC: 10.4.0.4

Redes aceleradas: **Deshabilitado**

Reglas de puerto de entrada Reglas de puerto de salida Grupos de seguridad de aplicación Equilibrio de carga

Grupo de seguridad de red M2018631591-4-nsg (se conectó a la interfaz de red: m2018631591-4449)

Impactos 0 subredes, 1 interfaces de red

Agregar regla de puerto de entrada

Prioridad	Nombre	Puerto	Protocolo	Origen	Destino	Acción
300	SSH	22	TCP	Cualquiera	Cualquiera	Permitir
65000	AllowVnetInBound	Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork	Permitir
65001	AllowAzureLoadBalancerInBound	Cualquiera	Cualquiera	AzureLoadBalancer	Cualquiera	Permitir
65500	DenyAllInBound	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Denegar

En el apartado de redes de cada una de las maquinas virtuales se le da clic en agregar regla de puerto de entrada para agregar el puerto 50000 que es el que use para el programa, el puerto 50000 se agrega de la siguiente manera:

Microsoft Azure

Inicio > M2018631591-0

M2018631591-0 | Redes

Máquina virtual

Buscar (Ctrl+/)

Adjuntar interfaz de red Desasociar interfaz de red

Configuración de IP [ⓘ]

ipconfig1 (Principal)

Interfaz de red: m2018631591-03 Reglas de seguridad vigentes

Red virtual/subred: M2018631591-0_group-vnet/default IP pública de NIC: M2018631591-0-ip IP privada de NIC: 10.4.0.4

Reglas de puerto de entrada Reglas de puerto de salida Grupos de seguridad de aplicación

Grupo de seguridad de red M2018631591-0-nsg (se conectó a la interfaz de red: m2018631591-03)

Impactos 0 subredes, 1 interfaces de red

Prioridad	Nombre	Puerto
300	SSH	22
310	Port_50000	Cualquiera
65000	AllowVnetInBound	Cualquiera
65001	AllowAzureLoadBalancerInBound	Cualquiera
65500	DenyAllInBound	Cualquiera

Port_50000

M2018631591-0-nsg

Guardar Descartar Eliminar

Origen [ⓘ]

Any

Intervalos de puertos de origen * [ⓘ]

*

Destino [ⓘ]

Any

Servicio [ⓘ]

Custom

Intervalos de puertos de destino * [ⓘ]

50000

Protocolo

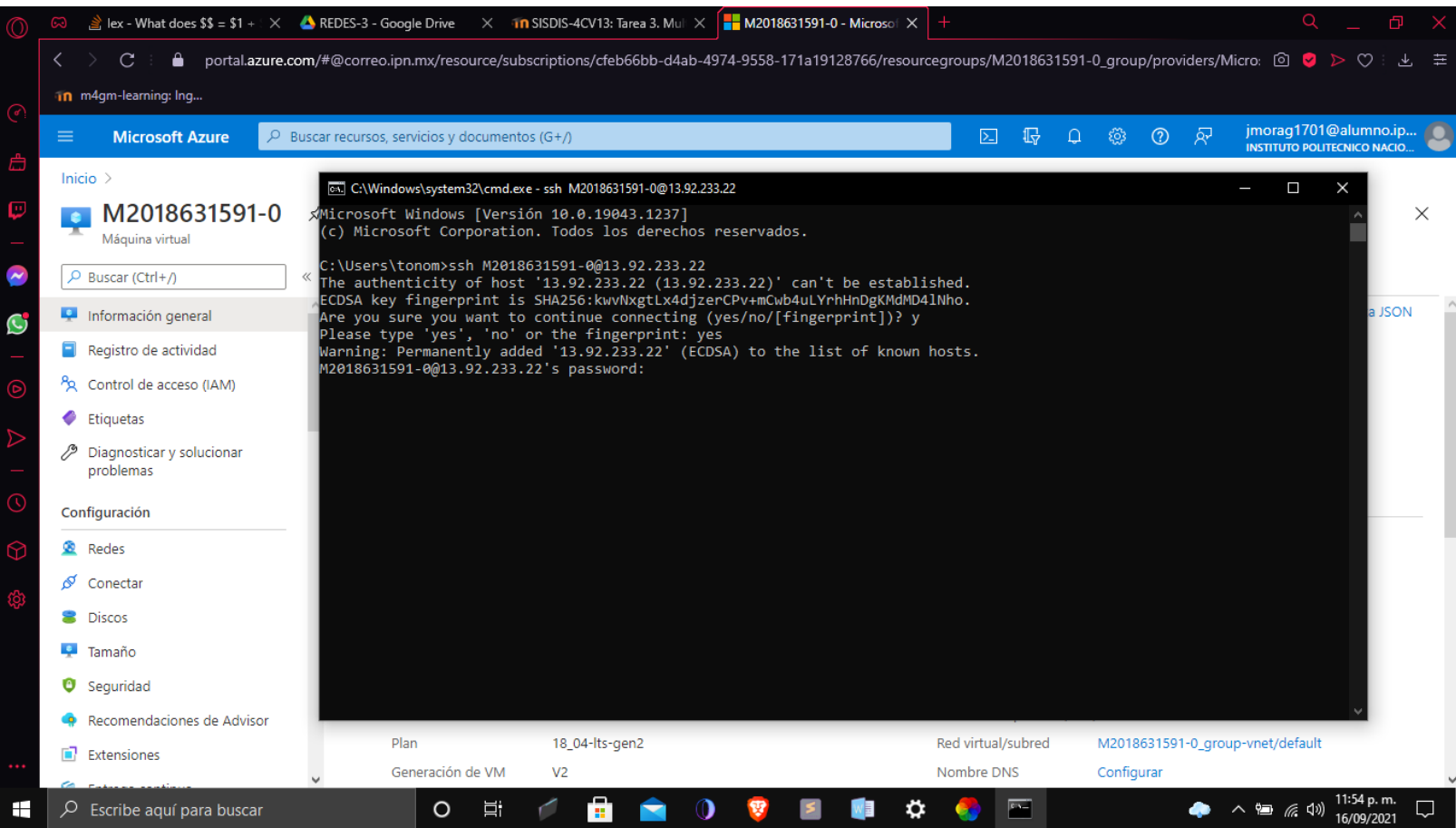
Any

☒ TCP

UDP

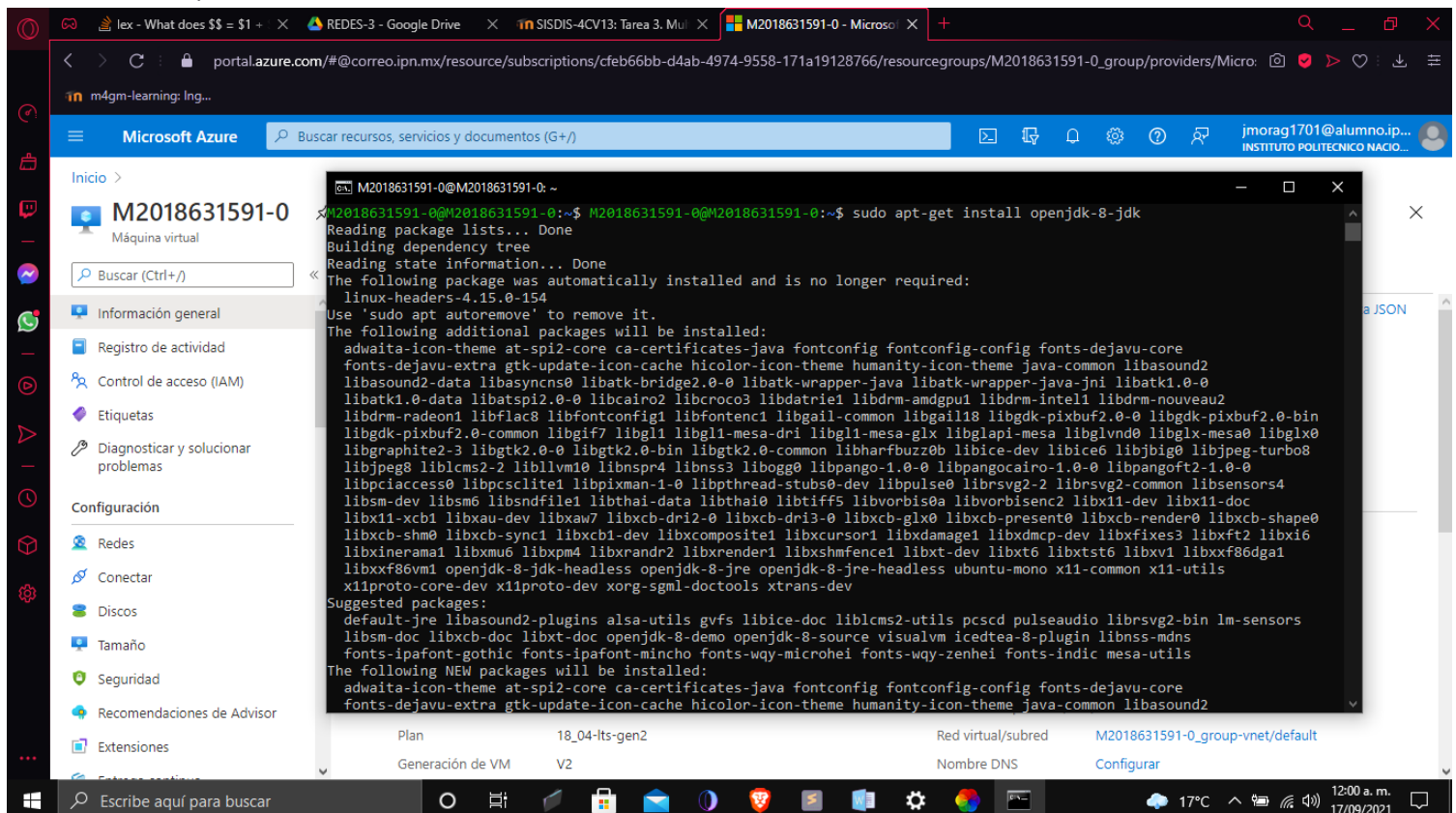
ICMP

Acción



En la captura anterior se muestra como por medio de nuestra consola de cmd de Windows entramos a nuestra maquina virtual por medio de ssh

Una vez dentro de la maquina virtual instale el jdk 8 que es lo que se muestra en la siguiente captura.



The screenshot shows a Windows desktop environment. In the background, a Sublime Text editor is open with a file named `MultMatDis.java`. The code is a Java program for matrix multiplication. It includes imports for `java.net.Socket`, `java.net.ServerSocket`, `java.io.DataOutputStream`, `java.io.DataInputStream`, and `java.lang.Thread`. The class `MultMatDis` contains static variables for matrix dimensions (`N=10`), indices (`i, j, k`), and a checksum. It also defines static matrices `A`, `B`, and `C`. A `Worker` class is defined as an inner class, extending `Thread`, with a `run` method that implements the matrix multiplication logic. The main method is not fully visible but appears to start a `Worker` thread.

In the foreground, a terminal window titled `GNU nano 2.9.3` is open, showing the same Java code being edited. The terminal prompt is `M2018631591-0@M2018631591-0: ~`. The code in the terminal matches the code in the Sublime Text editor.

En la captura anterior se muestra como por medio del comando “nano” coloque el código de mi programa para posteriormente guardarlo en la maquina virtual.

Y una vez guardado lo siguiente que hice fue compilar el programa

The screenshot shows a Windows desktop environment. In the background, a web browser is open to a Moodle assignment page. The page title is `m4gm-learning: Ing...`. The page content includes a header with the `m4gm` logo and a language selector set to `Español (México) (es_mx)`. The main content area shows the assignment details, including a title `Notar que en el paso 3 de...` and a description `Se deberá subir a la plata...`. The assignment status is `Sin intento` and the value is `20% (1.4)`.

In the foreground, a terminal window titled `M2018631591-0@M2018631591-0: ~` is open, showing the command `javac MultMatDis.java` being executed. The terminal prompt is `M2018631591-0@M2018631591-0: ~$`.

```
M2018631591-0@M2018631591-0: ~$ nano
M2018631591-0@M2018631591-0:~$ javac MultMatDis.java
M2018631591-0@M2018631591-0:~$ nano MultMatDis.java
M2018631591-0@M2018631591-0:~$ javac MultMatDis.java
M2018631591-0@M2018631591-0:~$ java MultMatDis 0
checksum= 186750
Matriz A
0 1 2 3 4 5 6 7 8 9
3 4 5 6 7 8 9 10 11 12
6 7 8 9 10 11 12 13 14 15
9 10 11 12 13 14 15 16 17 18
12 13 14 15 16 17 18 19 20 21
15 16 17 18 19 20 21 22 23 24
18 19 20 21 22 23 24 25 26 27
21 22 23 24 25 26 27 28 29 30
24 25 26 27 28 29 30 31 32 33
27 28 29 30 31 32 33 34 35 36
Matriz B
0 3 6 9 12 15 18 21 24 27
-1 2 5 8 11 14 17 20 23 26
-2 1 4 7 10 13 16 19 22 25
-3 0 3 6 9 12 15 18 21 24
-4 -1 2 5 8 11 14 17 20 23
-5 -2 1 4 7 10 13 16 19 22
-6 -3 0 3 6 9 12 15 18 21
-7 -4 -1 2 5 8 11 14 17 20
-8 -5 -2 1 4 7 10 13 16 19
-9 -6 -3 0 3 6 9 12 15 18
Matriz C
855 810 765 720 675 630 585 540 495 450
1260 1185 1110 1035 960 885 810 735 660 585
1665 1560 1455 1350 1245 1140 1035 930 825 720
2070 1935 1800 1665 1530 1395 1260 1125 990 855
2475 2310 2145 1980 1815 1650 1485 1320 1155 990
2880 2685 2490 2295 2100 1905 1710 1515 1320 1125
3285 3060 2835 2610 2385 2160 1935 1710 1485 1260
3690 3435 3180 2925 2670 2415 2160 1905 1650 1395
4095 3810 3525 3240 2955 2670 2385 2100 1815 1530
4500 4185 3870 3555 3240 2925 2610 2295 1980 1665
M2018631591-0@M2018631591-0:~$
```

Después de ejecutar todos los nodos, en nuestra maquina virtual M2018631591-0 se muestra la pantalla anterior cuando el valor de N es igual a 10

Y cuando el valor de N es igual a 1500 nos muestra la siguiente pantalla

The screenshot shows a web browser window with multiple tabs. The active tab is 'm4gm.com/moodle/mod/assign/view.php'. The page content includes a sidebar with a menu, a header with 'm4gm', and a main area titled 'Tarea'. The terminal window overlaid on the page shows the following commands and output:

```
M2018631591-0@M2018631591-0: ~$ java MultMatDis 0
checksum= 17065693406250000
M2018631591-0@M2018631591-0:~$
```

The browser window also shows a search bar at the bottom with the text 'Escribe aquí para buscar' and a system tray at the bottom right indicating the time as 03:44 p. m. on 17/09/2021.

CÓDIGO FUENTE DEL PROGRAMA

/*Tarea 3 Sistemas Distribuidos Realizada por:
JOSE ANTONIO MORA GUZMAN DEL GRUPO 4CV13*/

//BIBLIOTECAS A UTILIZAR

```
import java.net.Socket;
import java.net.ServerSocket;
import java.io.DataOutputStream;
import java.io.DataInputStream;
import java.lang.Thread;
class MultMatDis{
```

```
    static Object obj = new Object(); //OBJ SERA USADO EN EL LOCK PARA QUE ENTRE UN HILO A LA VEZ A LA
    ZONA CRITICA
```

```
    static int N = 10; //N que debe ser N=10 para desplegar A B C y CHECKSUM
```

```
        //N=1500 Imprime el checksum
```

```
    //Variables que se van a usar para los ciclos
```

```
    static int i=0;
```

```
    static int j=0;
```

```
    static int k=0;
```

```
    static long checksum = 0;
```

```
    static String ip ="10.0.0.4"; //IP DEL NODO 0(Servidor)
```

```
    static int port=50000; //PUERTO
```

```
    //MATRICES A USAR: A,B,C DE TIPO LONG de NxN
```

```
    static long[][] A = new long[N][N];
```

```
    static long[][] B = new long[N][N];
```

```
    static long[][] C = new long[N][N];
```

```
static class Worker extends Thread{//Clase PAra los Hilos
```

```
    Socket conexion;
```

```
    Worker(Socket conexion){
```

```
        this.conexion = conexion;
```

```
    }
```

```
    public void run(){ //Cuando Inicia un hilo entra al run
```

```
        try {
```

```
            //Dividimos A1 y A2 en (N/2)xN y B1 y B2 en Nx(N/2)
```

```
            long[][] A1 = new long[N/2][N];
```

```
            long[][] A2 = new long[N/2][N];
```

```
            //B1 y B2 en este momento son de [N/2][N] debido que con la transpuesta queda [N][N/2]
```

```
            long[][] B1 = new long[N/2][N];
```

```
            long[][] B2 = new long[N/2][N];
```

```
            DataInputStream entrada = new DataInputStream(conexion.getInputStream()); //Leer
```

```
            DataOutputStream salida = new DataOutputStream(conexion.getOutputStream()); //escribir
```

```
            //Recibimos el numero de nodo
```

```
            int nodos = entrada.readInt();
```

```
            // es parte del NODO 0
```

```
            //Vamos a mandar las matrices a cada nodo
```

```
            if (nodos == 1){
```

```
                for(i = 0; i < (N/2); i++){
```

```

        for(j = 0; j < N; j++){
            A1 [i][j] = A [i][j];
            B1 [i][j] = B [i][j];
            salida.writeLong(A1[i][j]); // 3 Enviar la matriz A1 al nodo 1
            salida.writeLong(B1[i][j]); // 4 Enviar la matriz B1 al nodo 1
        }
    }

} else if (nodos == 2){
    for(i = 0; i < (N/2); i++){
        for(j = 0; j < N; j++){
            A1 [i][j] = A [i][j];
            salida.writeLong(A1[i][j]); // 5 Enviar la matriz A1 al nodo 2.
        }
    }
    for(i = (N/2); i < N; i++){
        for(j = 0; j < N; j++){
            B2 [i - (N/2)][j] = B [i][j];
            salida.writeLong(B2[i - (N/2)][j]); // 6 Enviar la matriz B2 al nodo 2.
        }
    }
} else if (nodos == 3){
    for(i = (N/2); i < N; i++){
        for(j = 0; j < N; j++){
            A2 [i - (N/2)][j] = A [i][j];
            salida.writeLong(A2[i - (N/2)][j]); //7 Enviar la matriz A2 al nodo 3.
        }
    }
    for(i = 0; i < (N/2); i++){
        for(j = 0; j < N; j++){
            B1 [i][j] = B [i][j];
            salida.writeLong(B1[i][j]); // 8 Enviar la matriz B1 al nodo 3.
        }
    }
} else if (nodos == 4){
    for(i = (N/2); i < N; i++){
        for(j = 0; j < N; j++){
            A2 [i - (N/2)][j] = A [i][j];
            B2 [i - (N/2)][j] = B [i][j];
            salida.writeLong(A2[i - (N/2)][j]); // 9 Enviar la matriz A2 al nodo 4.
            salida.writeLong(B2[i - (N/2)][j]); // 10 Enviar la matriz B2 al nodo 4.
        }
    }
}

synchronized(obj){//Vamos a recibir las matrices en cada nodo todo esto
    //en synchronized para que no entre mas de un hilo a la vez
    if(nodos == 1){// 11 Recibe la matriz C1 del nodo 1
        long[][] C1 = new long[N/2][N/2];
        for(i = 0; i < (N/2); i++){
            for(j = 0; j < (N/2); j++){
                C1[i][j] = entrada.readLong(); //asigna a C1 lo que recibe en entrada.readLong()
            }
        }
    }
}

```

```

        C[i][j] = C1[i][j]; //a C le asigna lo que hay en C1
    }
}
}else if (nodos == 2){ // 12 Recibe la matriz C2 del nodo 2
    long[][] C2 = new long[N/2][N];
    for(i = 0; i < (N/2); i++){
        for(j = (N/2); j < N; j++){
            C2[i][j] = entrada.readLong();
            C[i][j] = C2[i][j];
        }
    }
}else if (nodos == 3){ // 13 Recibe la matriz C3 del nodo 3
    long[][] C3 = new long[N][N/2];
    for(i = (N/2); i < N; i++){
        for(j = 0; j < (N/2); j++){
            C3[i - (N/2)][j] = entrada.readLong();
            C[i][j] = C3[i - (N/2)][j];
        }
    }
}else if (nodos == 4){ // 14 Recibe la matriz C4 del nodo 4
    long[][] C4 = new long[N][N];
    for(i = (N/2); i < N; i++){
        for(j = (N/2); j < N; j++){
            C4[i - (N/2)][j - (N/2)] = entrada.readLong();
            C[i][j] = C4[i - (N/2)][j - (N/2)];
        }
    }
}

}

// Cerramos la conexion, la entrada y la salida
entrada.close();
salida.close();
conexion.close();
} catch (Exception e) {
    System.out.println(e.getMessage());
}
}
}

public static void main(String[] args) throws Exception {
    if (args.length != 1){ //Si no introduces parametros al ejecutar
        System.err.println("Debes poner como argumento el numero de nodo por ejemplo: java MultMatDis
<nodo>");
        System.exit(0);
    }
    int nodo = Integer.valueOf(args[0]);

    //***** NODO 0 *****

    if (nodo == 0){
        //MATRICES ORIGINALES DE NxN

```

```

A = new long[N][N];
B = new long[N][N];
C = new long[N][N];
//1 inicializamos nuestras matrices A y B
for (i = 0; i < N; i++){
    for (j = 0; j < N; j++){
        A[i][j] = 3*i+j;
        B[i][j] = 3*i-j;
    }
}
//2 transponer la matriz B
for (i = 0; i < N; i++){
    for (j = 0; j < i; j++){
        //intercambio de indices de B como en MultiplicaMatriz_2
        long x = B[i][j];
        B[i][j] = B[j][i];
        B[j][i] = x;
    }
}
ServerSocket servidor = new ServerSocket(50000); //creamos servidor puerto 50000
// vamos a aceptar 4 clientes (nodo 1,2,3,4)
Worker[] w = new Worker[4];

int numcliente = 0;
while (numcliente != 4){
    Socket conexion = servidor.accept();
    w[numcliente] = new Worker(conexion);
    w[numcliente].start();
    numcliente++;
}
// Esperamos a que se ejecute el hilo
int y = 0;
while (y != 4){
    w[y].join();
    y++;
}
// cerramos el servidor
servidor.close();
// 14. Calcular el checksum de la matriz C.
for( i = 0; i < N; i++){
    for( j = 0; j < N; j++){
        checksum += C[i][j];
    }
}
// 16 Desplegar el checksum de la matriz C.
System.out.println("checksum= " + checksum);
// 17 Si N=10 entonces desplegar las matrices A,B yC
if (N == 10){
    System.out.println("Matriz A ");
    imprimematriz(A,N,N);
    System.out.println("Matriz B ");
    imprimematriz(B,N,N);
    System.out.println("Matriz C ");
    imprimematriz(C,N,N);
}

```

```

    }
} else { // Si es algun otro nodo
    // matrices que usaremos para recibir
    long[][] tempA = new long[N/2][N];
    long[][] tempB = new long[N/2][N];
    // matriz que usaremos para A1 x B1
    long[][] resC = new long[N/2][N/2];

    //***** NODO 1 *****

    if (nodo == 1) {
        Socket conexion = new Socket(ip, port);
        DataInputStream entrada = new DataInputStream(conexion.getInputStream());
        DataOutputStream salida = new DataOutputStream(conexion.getOutputStream());
        // Enviamos al nodo 0 (Servidor) el nodo
        salida.writeInt(nodo);

        for (i = 0; i < (N/2); i++) {
            for (j = 0; j < N; j++) {
                tempA[i][j] = entrada.readLong(); //1 Recibir del nodo 0 la matriz A1.
                tempB[i][j] = entrada.readLong(); //2 Recibir del nodo 0 la matriz B1.
            }
        }
        for (i = 0; i < (N/2); i++) {
            for (j = 0; j < (N/2); j++) {
                for (k = 0; k < N; k++) {
                    resC[i][j] += tempA[i][k] * tempB[j][k]; //3 Realizar el producto C1=A1xB1 (Renglon por
renglon)
                }
            }
        }
        //4 Enviar la matriz C1 al nodo 0.
        for (i = 0; i < (N / 2); i++) {
            for (j = 0; j < (N / 2); j++) {
                salida.writeLong(resC[i][j]);
            }
        }
        //cerramos entrada, salida y la conexión
        entrada.close();
        salida.close();
        conexion.close();
    }

    //***** NODO 2 *****

    else if (nodo == 2) {
        Socket conexion = new Socket(ip, port);
        DataInputStream entrada = new DataInputStream(conexion.getInputStream());
        DataOutputStream salida = new DataOutputStream(conexion.getOutputStream());
        // Enviamos al nodo 0 (servidor) el nodo
        salida.writeInt(nodo);
        for (i = 0; i < (N/2); i++) {
            for (j = 0; j < N; j++) {

```

```

        tempA[i][j] = entrada.readLong();//1 Recibir del nodo 0 la matriz A1.
    }
}
for(i = (N/2); i < N; i++){
    for(j = 0; j < N; j++){
        tempB[i-(N/2)][j] = entrada.readLong();//2 Recibir del nodo 0 la matriz B2.

    }
}

for ( i = 0; i < (N/2); i++){
    for (j = 0; j < (N/2); j++){
        for (k = 0; k < N; k++){
            resC[i][j] += tempA[i][k] * tempB[j][k];//3 Realizar el producto C2=A1xB2 (Renglon por
renglon)
        }
    }
}

for( i = 0; i < (N / 2); i++){
    for( j = 0; j < (N / 2); j++){
        salida.writeLong(resC[i][j]);//4 Enviar la matriz C2 al nodo 0.
    }
}
entrada.close();
salida.close();
conexion.close();
}

//***** NODO 3 *****

else if (nodo == 3){
    Socket conexion = new Socket(ip, port);
    DataInputStream entrada = new DataInputStream(conexion.getInputStream());
    DataOutputStream salida = new DataOutputStream(conexion.getOutputStream());
    salida.writeInt(nodo);

    for( i = (N/2); i < N; i++){
        for( j = 0; j < N; j++){
            tempA[i-(N/2)][j] = entrada.readLong();//1 Recibir del nodo 0 la matriz A2
        }
    }
    for( i = 0; i < (N/2); i++){
        for( j = 0; j < N; j++){
            tempB[i][j] = entrada.readLong();//2 Recibir del nodo 0 la matriz B1.
        }
    }

    for ( i = 0; i < (N/2); i++){
        for (j = 0; j < (N/2); j++){
            for (k = 0; k < N; k++){
                resC[i][j] += tempA[i][k] * tempB[j][k];//3 Realizar el producto C3=A2xB1 (renglon por
renglon)
            }
        }
    }
}

```



```

    }
}
}
for(i = 0; i < (N/2); i++){
    for(j = 0; j < (N/2); j++){
        salida.writeLong(resC[i][j]); //4 Enviar la matriz C3 al nodo 0.
    }
}
entrada.close();
salida.close();
conexion.close();
}

//***** NODO 4 *****

else if (nodo == 4){
    Socket conexion = new Socket(ip, port);
    DataInputStream entrada = new DataInputStream(conexion.getInputStream());
    DataOutputStream salida = new DataOutputStream(conexion.getOutputStream());
    salida.writeInt(nodo);

    for(i = (N/2); i < N; i++){
        for(j = 0; j < N; j++){
            tempA[i-(N/2)][j] = entrada.readLong(); //1 Recibir del nodo 0 la matriz A2
            tempB[i-(N/2)][j] = entrada.readLong(); //2 Recibir del nodo 0 la matriz B2.
        }
    }

    for ( i = 0; i < (N/2); i++){
        for (j = 0; j < (N/2); j++){
            for (k = 0; k < N; k++){
                resC[i][j] += tempA[i][k] * tempB[j][k]; //3 Realizar el producto C4=A2xB2(renglon por
renglon)
            }
        }
    }

    for( i = 0; i < (N/2); i++){
        for( j = 0; j < (N/2); j++){
            salida.writeLong(resC[i][j]); //4 Enviar la matriz C4 al nodo 0.
        }
    }
    entrada.close();
    salida.close();
    conexion.close();
}

}
}

// Clase para Imprimir Las matrices
private static void imprimematriz(long[][] mat, long filas, long columnas) {
    for (i = 0; i < filas; i++){

```

```
        for (j = 0; j < columnas; j++){  
            System.out.print(mat[i][j] + " ");  
        }  
        System.out.println("");  
    }  
}  
}
```

CONCLUSIONES

Esta practica fue mas complicada que las anteriores pero prácticamente traía de todo lo que habíamos visto durante el parcial, lo complicado fue el hacer que funcionaran en azure debido a que primero al ser la maquina de 1gb de ram no era tan rápida y la conexión se quedaba esperando pero al cambiar eso ya el programa por fin pudo ejecutarse de manera correcta.