



**INSTITUTO POLITÉCNICO NACIONAL.  
ESCUELA SUPERIOR DE CÓMPUTO.**



## **Sistemas Distribuidos**

**Tarea 03: Reporte de “Multiplicación de matrices  
distribuida utilizando paso de mensajes”**

Alumno: Oaxaca Pérez David Arturo

Grupo:  
4CV12

A cargo del profesor:  
PINEDA GUERRERO CARLOS

## Contenido

|  |    |
|--|----|
| Introducción .....   | 3  |
| Desarrollo .....   | 4  |
| Creación de las máquinas virtuales.....                              | 6  |
| Configuración de las máquinas virtuales .....                        | 11 |
| Compilación y ejecución del programa .....                           | 15 |
| Resultado de la ejecución del programa para N = 10 .....             | 18 |
| Resultado de la ejecución del programa para N = 1500 .....           | 20 |
| Conclusiones .....   | 23 |
| Anexos.....  | 24 |
| Código .....   | 24 |
| Creación de la quinta máquina virtual (Cambio a 3.5 GB de RAM) ..... | 32 |

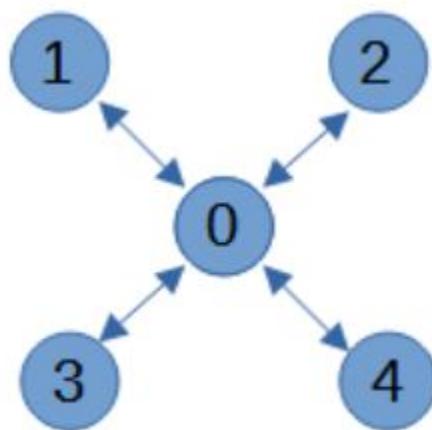
## Introducción

Para esta tarea se hará uso de varios de los conceptos vistos anteriormente en clase, entre ellos, parte de la actividad para multiplicar matrices que consiste en transponer la matriz B, para que la multiplicación de  $A * B$  sea más rápida, además de eso, el cálculo se hará de manera distribuida, es decir, cada matriz se partirá en 2 para realizar las multiplicaciones y posteriormente se unirá para tener una matriz C completa.

Este concepto se puede ver más claramente en las imágenes proporcionadas por la misma asignación como puede ser esta:



En cuanto a la parte distribuida del programa, esta se hará usando 5 máquinas virtuales proporcionadas por Azure, donde se le proporcionara a los programas un ID de nodo y trabajaran para calcular la multiplicación de matrices de forma distribuida mediante la siguiente topología:



## Desarrollo

Para el desarrollo de esta práctica primero desarrollaremos lo que es el programa y lo probaremos de manera local, para posteriormente crear 5 máquinas virtuales en Azure y ejecutarlo en cada una de ellas para que el programa funcione de manera distribuida.

Las instrucciones que realizará cada nodo, declaradas previamente en la asignación de la página también se incluirán en esta parte:

### Nodo 0:

- 1.- Inicializar matrices A y B
- 2.- Transponer la matriz B
- 3.- Enviar la matriz A1 al nodo 1
- 4.- Enviar la matriz B1 al nodo 1
- 5.- Enviar la matriz A1 al nodo 2
- 6.- Enviar la matriz A2 al nodo 2
- 7.- Enviar la matriz A2 al nodo 3
- 8.- Enviar la matriz B1 al nodo 3
- 9.- Enviar la matriz A2 al nodo 4
- 10.- Enviar la matriz B2 al nodo 4
- 11.- Recibir la matriz C1 del nodo 1
- 12.- Recibir la matriz C2 del nodo 2
- 13.- Recibir la matriz C3 del nodo 3
- 14.- Recibir la matriz C4 del nodo 4
- 15.- Calcular el checksum de la matriz C
- 16.- Desplegar el checksum de la matriz C
- 17.- Si  $N = 10$  entonces desplegar las matrices A, B y C

### Nodo 1:

- 1.- Recibir del nodo 0 la matriz A1
- 2.- Recibir del nodo 0 la matriz B1
- 3.- Realizar el producto  $C1 = A1 \times B1$  (renglón por renglón)
- 4.- Enviar la matriz C1 al nodo 0

**Nodo 2:**

- 1.- Recibir del nodo 0 la matriz A1
- 2.- Recibir del nodo 0 la matriz B2
- 3.- Realizar el producto  $C2 = A1 \times B2$  (renglón por renglón)
- 4.- Enviar la matriz C2 al nodo 0

**Nodo 3:**

- 1.- Recibir del nodo 0 la matriz A2
- 2.- Recibir del nodo 0 la matriz B1
- 3.- Realizar el producto  $C3 = A2 \times B1$  (renglón por renglón)
- 4.- Enviar la matriz C3 al nodo 0

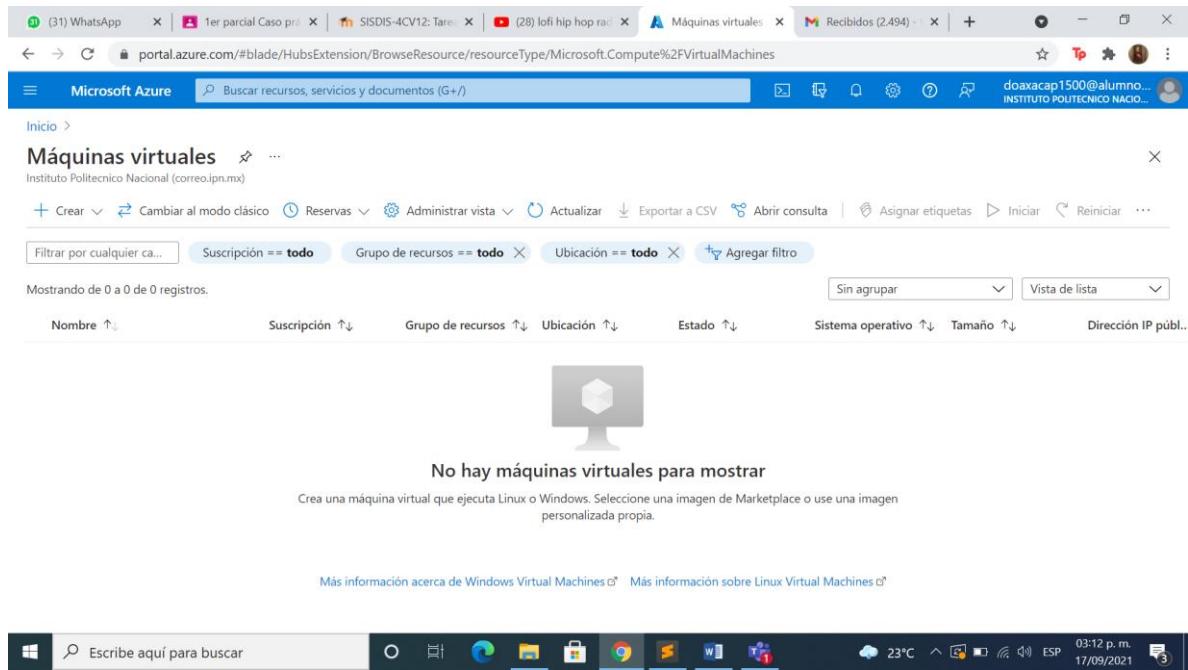
**Nodo 4:**

- 1.- Recibir del nodo 0 la matriz A2
- 2.- Recibir del nodo 0 la matriz B2
- 3.- Realizar el producto  $C4 = A2 \times B2$  (renglón por renglón)
- 4.- Enviar la matriz C4 al nodo 0

## Creación de las máquinas virtuales

En esta parte de la práctica solo se detallará la creación de una máquina virtual, a pesar de que se usen 5 para no alargar el desarrollo con procedimientos que puedan parecer redundantes posteriormente.

En esta primera captura podemos observar que no tenemos ninguna máquina virtual, apenas vamos a crear una dando click en el botón de crear.



Las máquinas virtuales que se usarán serán Ubuntu 18.04 LTS – Gen 2 pues es más sencillo instalar y utilizar Java en ellas que en una de Windows donde lo mejor sería usar una plantilla que ya incluya JDK pero que podría salir más costosa o general podría ser un procedimiento más tardado.

En esta siguiente captura podemos ver los datos básicos para la creación de una nueva máquina virtual luego de dar click en el botón + Crear. Aquí escogemos el grupo de recursos creado para algunas de las actividades realizadas en clase llamado prueba\_group, nombramos a la máquina virtual como Prueba escogemos como región el Este de EE.UU. para tener una mayor disponibilidad de tamaños a elegir.

En esta captura podemos ver como escogemos una Imagen de un Ubuntu Server 18.04 LTS – Gen1 para la máquina virtual junto con un tamaño de 1 Gb de memoria. En el caso de la última máquina virtual escogemos un espacio de 3.5 GB de la familia D pues Azure tiene un límite de 4 máquinas virtuales con el mismo plan, fuera de eso es el mismo procedimiento.

Posteriormente en esta captura de pantalla podemos ver que para la cuenta de administrador escogemos como tipo de autenticación el uso de contraseña, escogemos como nombre de usuario Ubuntu y escribimos una contraseña.

Cuenta de administrador

Tipo de autenticación  Clave pública SSH  Contraseña

Nombre de usuario \*

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

Seleccionar puertos de entrada \*

**Revisar y crear** < Anterior Siguiente: Discos >

Dejamos los puertos para SSH (El 22) abiertos y posteriormente abriremos otros para la conexión por protocolo TCP que usaremos mediante los sockets.

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

Seleccionar puertos de entrada \*

**Este permitirá que todas las direcciones IP accedan a la máquina virtual.**  
Esto solo se recomienda para las pruebas. Use los controles avanzados de la pestaña Redes a fin de crear reglas para limitar el tráfico entrante a las direcciones IP conocidas.

**Revisar y crear** < Anterior Siguiente: Discos >

En las opciones de disco duro escogemos la opción de HDD estándar en el tipo de disco del sistema operativo para ahorrar costos y en administración deshabilitamos el diagnóstico de arranque.

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)

El tamaño de la máquina virtual seleccionada es compatible con los discos premium. Se recomienda SSD Premium para elevadas cargas de trabajo de E/S por segundo. Las máquinas virtuales con discos SSD Premium optan al acuerdo de nivel de servicio de conectividad del 99,9%.

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).

**Revisar y crear** < Anterior Siguiente: Redes >

En esta captura podemos ver que pasamos la validación al precionar “Revisar y crear” y ya podemos iniciar su creación.

Validación superada

Datos básicos Discos Redes Administración Opciones avanzadas Etiquetas Revisar y crear

PRODUCT DETAILS

B1s standard by Microsoft  Subscription credits apply  0,2194 MXN/hr [Pricing for other VM sizes](#)

TERMS

By clicking "Crear", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not

**Crear** < Anterior Siguiente > Descargar una plantilla para la automatización

En esta captura podemos ver que la implementación de la máquina está en curso.

Microsoft Azure | Buscar recursos, servicios y documentos (G+/-)

Inicio > CreateVm-Canonical.UbuntuServer-18.04-LTS-20210917151335 | Información general

Implementación

Buscar (Ctrl+F)

Eliminar Cancelar Volver a implementar Actualizar

Nos encantaría recibir sus comentarios. →

... Implementación en curso...

Se está realizando la implementación en el grupo de recursos 'grupo\_prueba'.

Nombre de implementación: CreateVm-Canonical.UbuntuServer-18... Hora de inicio: 17/9/2021 15:16:53

Suscripción: Azure para estudiantes Id. de correlación: 3df54963-c61a-475c-84f0-6369b4...

Grupo de recursos: grupo\_prueba

Detalles de implementación (Descargar)

| Recurso                  | Tipo | Estado | Detalles de la operación |
|--------------------------|------|--------|--------------------------|
| No hay ningún resultado. |      |        |                          |

Security Center  
Proteja sus aplicaciones e  
Vaya a Azure Security Cent

Tutoriales gratuitos de Mi  
Comience a aprender hoy

Trabajar con un experto  
Los expertos de Azure son  
proveedores de servicios c  
a administrar sus recursos  
primera línea de soporte t

Escribe aquí para buscar

23°C 03:17 p. m. ESP 17/09/2021

Y aquí finalmente podemos ver que se ha completado la implementación y podemos acceder al recurso.

Microsoft Azure | Buscar recursos, servicios y documentos (G+/-)

Inicio > CreateVm-Canonical.UbuntuServer-18.04-LTS-20210917151335 | Información general

Implementación

Buscar (Ctrl+F)

Eliminar Cancelar Volver a implementar Actualizar

Nos encantaría recibir sus comentarios. →

✓ Se completó la implementación

Nombre de implementación: CreateVm-Canonical.UbuntuServer-18... Hora de inicio: 17/9/2021 15:16:53

Suscripción: Azure para estudiantes Id. de correlación: 3df54963-c61a-475c-84f0-6369b4...

Grupo de recursos: grupo\_prueba

Detalles de implementación (Descargar)

Pasos siguientes

- Configurar el apagado automático Recomendado
- Supervisar el estado, el rendimiento y las dependencias de red de la máquina virtual Recomendado
- Ejecutar un script dentro de la máquina virtual Recomendado

Ir al recurso Crear otra VM

Security Center  
Proteja sus aplicaciones e  
Vaya a Azure Security Cent

Tutoriales gratuitos de Mi  
Comience a aprender hoy

Trabajar con un experto  
Los expertos de Azure son  
proveedores de servicios c  
a administrar sus recursos  
primera línea de soporte t

Escribe aquí para buscar

23°C 03:18 p. m. ESP 17/09/2021

Aquí accedemos al recurso y cómo podemos ver ya nos ha creado una máquina virtual, que será donde corramos el nodo 0. Podemos ver que su dirección IP es la 104.211.54.203 que le pasaremos a los demás nodos posteriormente.

The screenshot shows the Microsoft Azure portal interface. In the center, there's a detailed view of a VM named 'Prueba'. The 'Información general' section shows the VM is running in the 'grupo\_prueba' resource group, using a standard B1s size with 1 vCPU and 1 GB of memory. It's located in the United States (East US). The 'Sistema operativo' is listed as Linux (ubuntu 18.04). The 'Dirección IP pública' is clearly visible as 104.211.54.203. On the left sidebar, there are sections for 'Información general', 'Configuración' (Redes, Conectar, Discos, Tamaño, Seguridad), and 'Recomendaciones de Advisor'. At the bottom, there are tabs for 'Propiedades', 'Supervisión', 'Funcionalidades (7)', 'Recomendaciones', and 'Tutoriales'. A 'Redes' tab is selected, showing the public IP address 104.211.54.203. The Windows taskbar at the bottom shows the date and time as 17/09/2021, 03:19 p.m.

## Configuración de las máquinas virtuales

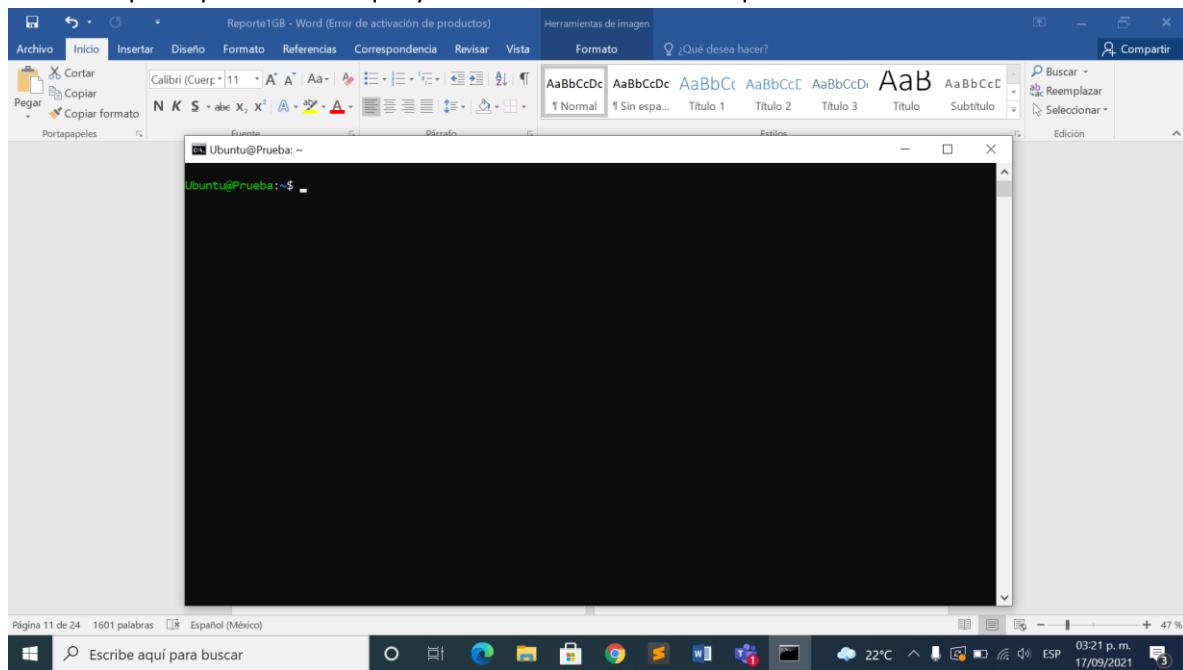
El primer paso es ingresar por medio de SSH en la línea de comando de Windows, escribimos:

ssh [Ubuntu@104.211.54.203](https://104.211.54.203)

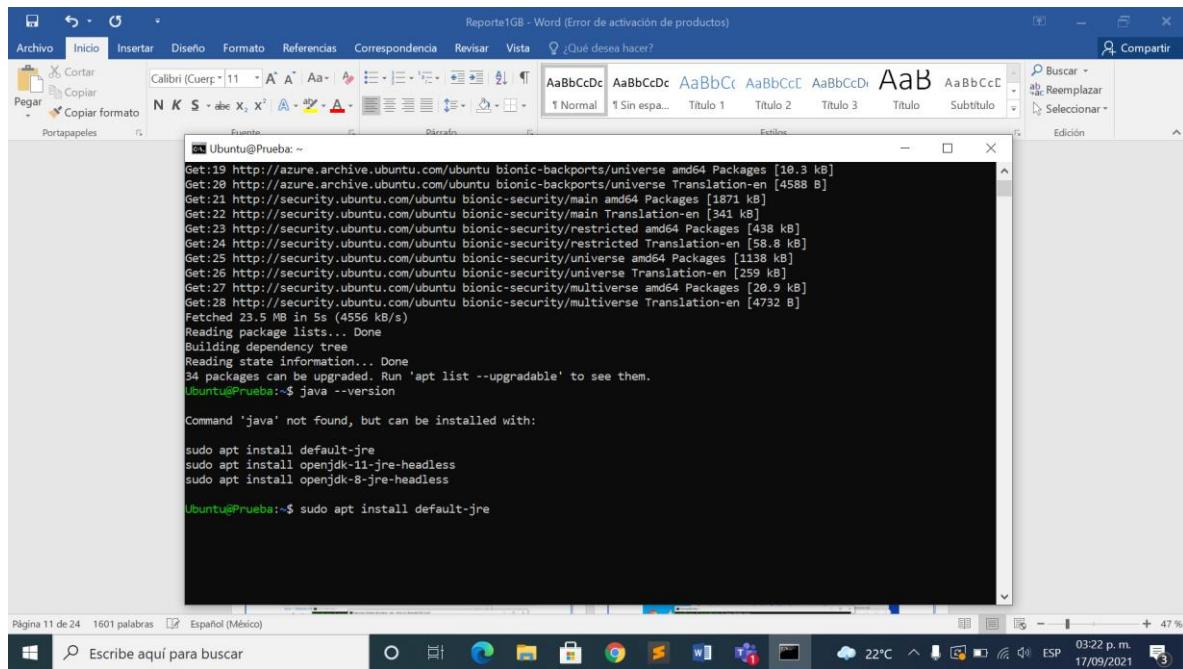
Y posteriormente introducimos la contraseña que escribimos en Azure como método de autenticación.

The screenshot shows a Microsoft Word document titled 'Reporte1GB - Word (Error de activación de productos)'. An embedded terminal window is open, showing the command 'ssh Ubuntu@104.211.54.203' being run. The terminal output shows the host key fingerprint and asks if the user wants to continue connecting. The user responds with 'yes'. It then adds the host to the known hosts file and prompts for the password. Once connected, it shows the Ubuntu 18.04 LTS login screen. The terminal then displays system information, including the load average (0.1), memory usage (20%), and swap usage (0%). It also shows the number of processes (118) and users logged in (0). The terminal concludes with a note about the programs included being free software and a warning about the lack of warranty. The Windows taskbar at the bottom shows the date and time as 17/09/2021, 03:20 p.m.

En esta captura podemos ver que ya tenemos acceso a la máquina virtual.



En esta captura aplicamos el comando de **sudo apt update** para posteriormente instalar el JDK y JRE, como se ve en la siguiente captura.



Reporte1GB - Word (Error de activación de productos)

```

Ubuntu@Prueba: ~
Adding debian:Go_Daddy_Root_Certificate_Authority_-_G2.pem
Adding debian:GlobalSign_ECC_Root_CA_-_R4.pem
Adding debian:Chambers_of_Commerce_Root_-_2008.pem
Adding debian:Microsec_e-Sign_Root_CA_2009.pem
Adding debian:Secure_Global_CA.pem
Adding debian:Amazon_Root_CA_3.pem
Adding debian:emSign_ECC_Root_CA_-_C3.pem
Adding debian:Autoridad_de_Certificación_Firmaprofesional_CIF_A62534068.pem
Adding debian:emSign_Root_CA_-_C1.pem
Adding debian:Hellenic_Academic_and_Research_Institutions_ECC_RootCA_2015.pem
Adding debian:SSL_com_EV_Root_Certification_Authority_ECC.pem
Adding debian:Microsoft_RSA_Root_Certificate_Authority_2017.pem
Adding debian:emSign_ECC_Root_CA_-_G3.pem
done.
Setting up default-jre (2:1.11-68ubuntu1~18.04.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.4) ...
Processing triggers for system (2.37-3ubuntu0.51) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ca-certificates (20210119~18.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

done.
done.
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Ubuntu@Prueba:~$ sudo apt install default-jdk

```

Página 13 de 23 1601 palabras Espanol (México)

Windows taskbar: Escribe aquí para buscar, Start button, Taskbar icons, Weather (22°C), Date (17/09/2021), Time (03:26 p. m.), Battery icon.

Para transferir nuestro archivo de java usaremos **sftp** y el comando **put**, en esta captura podemos ver como se usó para la segunda máquina virtual creada, ya que en la primera se hicieron algunas pruebas previas, en esta el procedimiento es mucho más claro.

Reporte1GB - Word (Error de activación de productos)

```

Símbolo del sistema - sftp -oPort=22 Ubuntu2@20.85.227.49

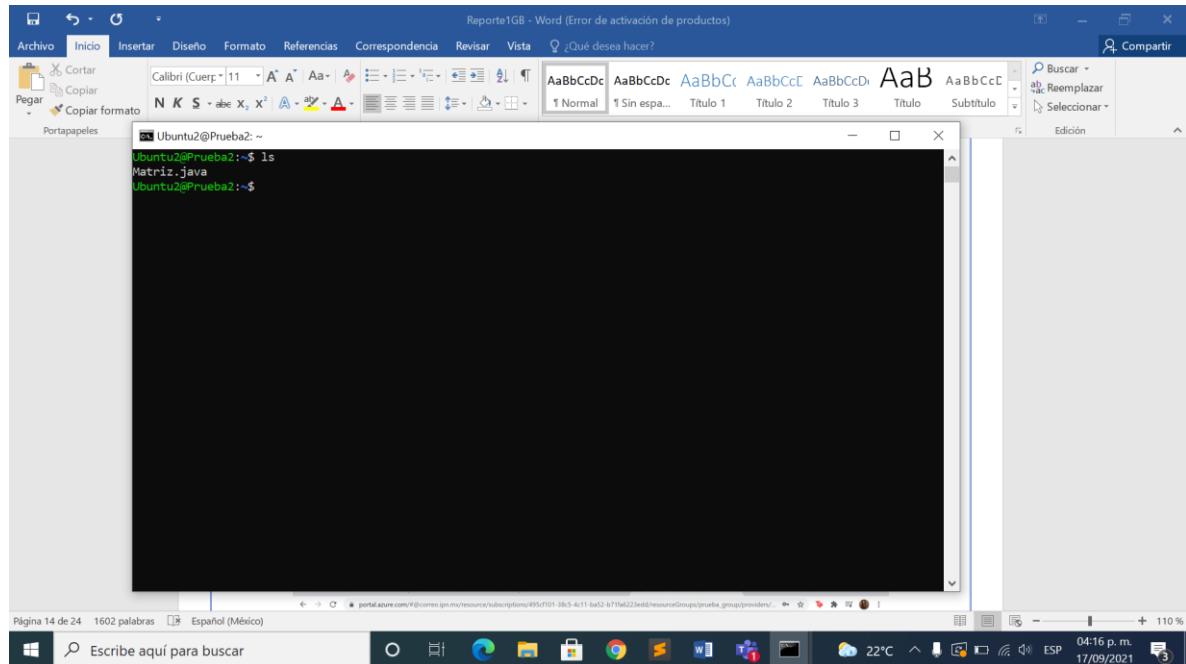
C:\Users\tdwda>sftp -oPort=22 Ubuntu2@20.85.227.49
Ubuntu2@20.85.227.49's password:
Connected to 20.85.227.49.
sftp> put C:\Users\tdwda\OneDrive\Escritorio\Tarea3\Matriz.java
Uploading C:/Users/tdwda/OneDrive/Escritorio/Tarea3/Matriz.java to /home/Ubuntu2/Matriz.java
          100% 8548   125.6KB/s  00:00
sftp>

```

Página 13 de 24 1602 palabras Espanol (México)

Windows taskbar: Escribe aquí para buscar, Start button, Taskbar icons, Weather (22°C), Date (17/09/2021), Time (04:16 p. m.), Battery icon.

En esta captura podemos ver que en efecto, ya fue transferido el archivo y usando el comando ls lo podemos visualizar.



Agregamos una regla de seguridad de entrada para permitir peticiones del protocolo TCP por el puerto 20000, esto se hace en las 5 máquinas virtuales.

The screenshot shows the Microsoft Azure portal interface. On the left, the navigation menu is expanded under the "Redes" section. In the center, a modal dialog titled "Agregar regla de seguridad de entrada" is open, showing the configuration for a new rule. The "Destino" dropdown is set to "Any", the "Intervalos de puertos de destino" dropdown is set to "20000", and the "Protocolo" radio button is selected for "TCP". The "Agregar" button is visible at the bottom right of the dialog. The taskbar at the bottom shows the Azure portal and other system icons.

En esta máquina virtual podemos ver el nombre que le ponemos a la regla de entrada.

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar for 'Prueba | Redes'. The main area is titled 'Agregar regla de seguridad de entrada' (Add Network Security Rule) for 'Prueba-nsg'. The configuration details are as follows:

- Protocolo: UDP
- Acción: Permitir
- Prioridad: 310
- Nombre: Port\_20000
- Descripción: (empty)

Below this, a table lists existing security rules:

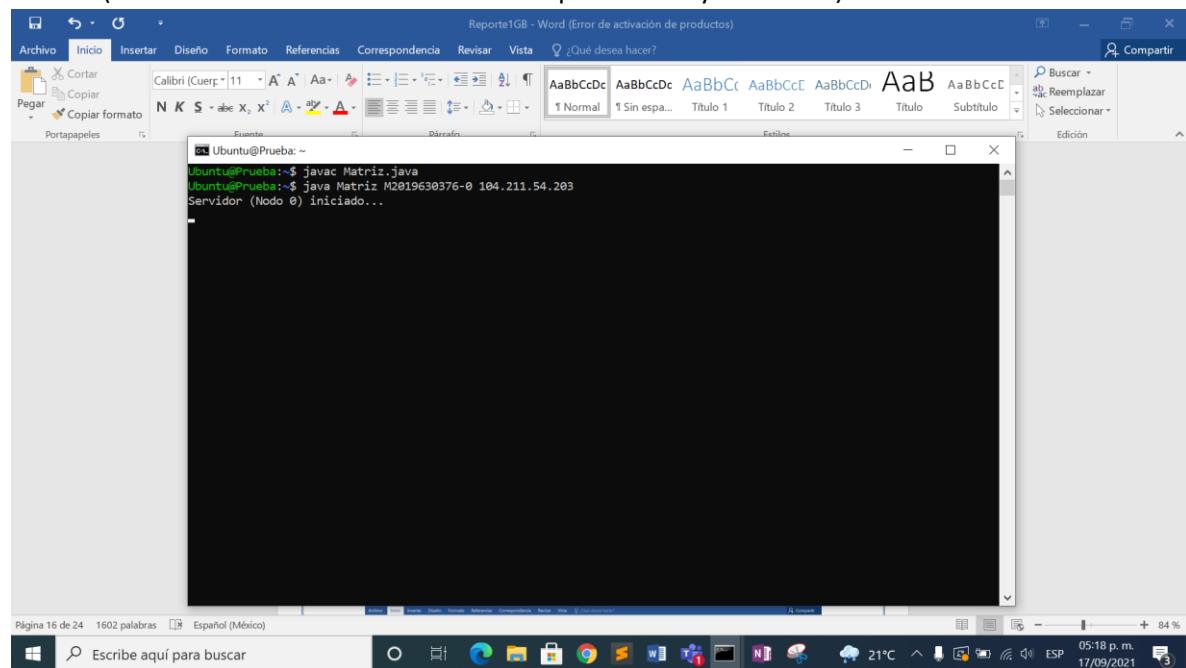
| Prioridad | Nombre                        | Puerto    |
|-----------|-------------------------------|-----------|
| 300       | SSH                           | 22        |
| 65000     | AllowVnetInBound              | Cualquier |
| 65001     | AllowAzureLoadBalancerInBound | Cualquier |
| 65500     | DenyAllInBound                | Cualquier |

## Compilación y ejecución del programa

En esta captura podemos ver como se compila el programa, sin errores.

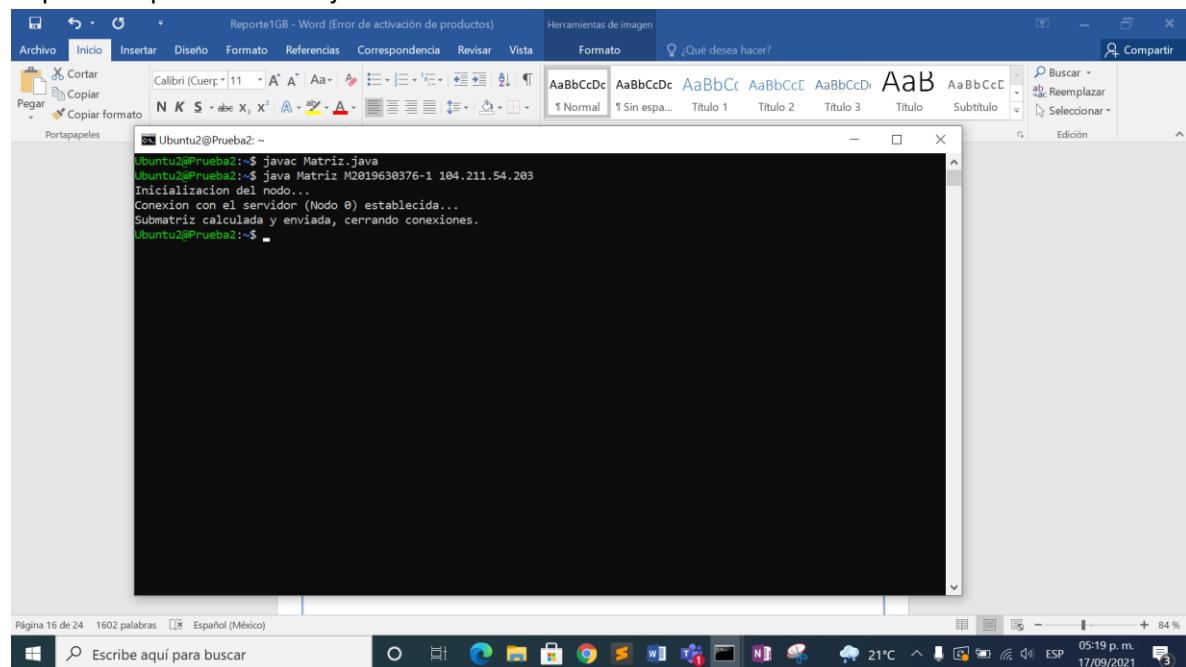
The screenshot shows a Microsoft Word document titled 'Reporte1GB - Word (Error de activación de productos)'. Inside the document, there is a screenshot of a terminal window from an Ubuntu system. The terminal shows the command 'javac Matriz.java' being entered and executed successfully, with no errors displayed.

Y posterior a eso lo ejecutamos, mandando como argumentos el ID de nodo y la dirección IP del nodo 0 (Incluso si de este mismo se trata aunque no se vaya a utilizar)



Las siguientes capturas son la compilación y ejecución del programa como diferentes nodos en distintas máquinas virtuales. Los tiempos pueden distanciarse algunos minutos porque a esa hora también me encontraba en otro trabajo escolar.

#### Captura de pantalla de la ejecución del nodo 1



## Captura de pantalla de la ejecución del nodo 2

A screenshot of a Microsoft Word document titled "Reporte1GB - Word (Error de activación de productos)". The document contains a black terminal window displaying the execution of a Java program. The terminal output shows:

```
Ubuntu3@Prueba3:~$ javac Matriz.java
Ubuntu3@Prueba3:~$ java Matriz M2019630376-2 104.211.54.203
Inicialización del nodo...
Conexión con el servidor (Nodo 0) establecida...
Submatriz calculada y enviada, cerrando conexiones.
Ubuntu3@Prueba3:~$
```

The Word document interface includes a ribbon bar with tabs like Archivo, Inicio, Insertar, Diseño, Formato, Referencias, Correspondencia, Revisar, Vista, and Compartir. The status bar at the bottom shows "Página 17 de 24" and "1602 palabras". The taskbar at the bottom of the screen shows various application icons.

## Captura de pantalla de la ejecución del nodo 3

A screenshot of a Microsoft Word document titled "Reporte1GB - Word (Error de activación de productos)". The document contains a black terminal window displaying the execution of a Java program. The terminal output shows:

```
Ubuntu4@Prueba4:~$ javac Matriz.java
Ubuntu4@Prueba4:~$ java Matriz M2019630376-3 104.211.54.203
Inicialización del nodo...
Conexión con el servidor (Nodo 0) establecida...
Submatriz calculada y enviada, cerrando conexiones.
Ubuntu4@Prueba4:~$
```

The Word document interface includes a ribbon bar with tabs like Archivo, Inicio, Insertar, Diseño, Formato, Referencias, Correspondencia, Revisar, Vista, and Compartir. The status bar at the bottom shows "Página 17 de 24" and "1602 palabras". The taskbar at the bottom of the screen shows various application icons.

## Captura de pantalla de la ejecución del nodo 4

The screenshot shows a Microsoft Word document titled "Reporte1GB - Word (Error de activación de productos)". A black terminal window is embedded within the document, displaying the following command-line session:

```
Ubuntu5@Prueba:~$ javac Matriz.java
Ubuntu5@Prueba:~$ java Matriz M2019630376-4 104.211.54.203
Inicialización del nodo...
Conexión con el servidor (Nodo 0) establecida...
Submatriz calculada y enviada, cerrando conexiones.
Ubuntu5@Prueba:~$
```

The Word document interface includes a ribbon bar with tabs like Archivo, Inicio, Insertar, Diseño, Formato, Referencias, Correspondencia, Revisar, Vista, and Compartir. The status bar at the bottom indicates "Página 18 de 24" and "1602 palabras".

## Resultado de la ejecución del programa para N = 10

Aquí podemos ver finalmente que el nodo 0 despliega el checksum de la matriz C el cual es **77250** y al ser N = 10 posteriormente despliega las matrices A, B y C

The screenshot shows a Microsoft Word document titled "Reporte1GB - Word (Error de activación de productos)". A black terminal window is embedded within the document, displaying the following command-line session:

```
Ubuntu5@Prueba:~$ ls
'Matriz$Worker.class'  Matriz.class  Matriz.java
Ubuntu5@Prueba:~$ clear
Ubuntu5@Prueba:~$ javac Matriz.java
Ubuntu5@Prueba:~$ java Matriz M2019630376-0 104.211.54.203
Servidor (Nodo 0) iniciado...
Nodo conectado, iniciando cálculo de multiplicación de matrices...

Checksum de la matriz C: 77250

Matriz A:
  0   1   2   3   4   5   6   7   8   9
  2   3   4   5   6   7   8   9   10  11
  4   5   6   7   8   9   10  11  12  13
  6   7   8   9   10  11  12  13  14  15
  8   9   10  11  12  13  14  15  16  17
  10  11  12  13  14  15  16  17  18  19
  12  13  14  15  16  17  18  19  20  21
  14  15  16  17  18  19  20  21  22  23
  16  17  18  19  20  21  22  23  24  25
  18  19  20  21  22  23  24  25  26  27

Matriz B:
  0   -1  -2  -3  -4  -5  -6  -7  -8  -9
```

The Word document interface includes a ribbon bar with tabs like Archivo, Inicio, Insertar, Diseño, Formato, Referencias, Correspondencia, Revisar, Vista, and Compartir. The status bar at the bottom indicates "Página 18 de 24" and "1602 palabras".

Aquí podemos ver cómo está desplegando la matriz A y B.

The screenshot shows a Microsoft Word document window titled "Reporte1GB - Word (Error de activación de productos)". Inside the document, there are two terminal windows from an Ubuntu system. The first terminal window displays the matrix A:

```
Checksum de la matriz C: 77250
Matriz A:
0 1 2 3 4 5 6 7 8 9
2 3 4 5 6 7 8 9 10 11
4 5 6 7 8 9 10 11 12 13
6 7 8 9 10 11 12 13 14 15
8 9 10 11 12 13 14 15 16 17
10 11 12 13 14 15 16 17 18 19
12 13 14 15 16 17 18 19 20 21
14 15 16 17 18 19 20 21 22 23
16 17 18 19 20 21 22 23 24 25
18 19 20 21 22 23 24 25 26 27
```

The second terminal window displays the matrix B:

```
0 -1 -2 -3 -4 -5 -6 -7 -8 -9
2 1 0 -1 -2 -3 -4 -5 -6 -7
4 3 2 1 0 -1 -2 -3 -4 -5
6 5 4 3 2 1 0 -1 -2 -3
8 7 6 5 4 3 2 1 0 -1
10 9 8 7 6 5 4 3 2 1
12 11 10 9 8 7 6 5 4 3
14 13 12 11 10 9 8 7 6 5
16 15 14 13 12 11 10 9 8 7
18 17 16 15 14 13 12 11 10 9
```

Aquí podemos ver como está desplegando la matriz C:

The screenshot shows a Microsoft Word document window titled "Reporte1GB - Word (Error de activación de productos)". Inside the document, there is one terminal window from an Ubuntu system. The terminal window displays the matrix C:

```
Checksum de la matriz C: 77250
Matriz B:
0 -1 -2 -3 -4 -5 -6 -7 -8 -9
2 1 0 -1 -2 -3 -4 -5 -6 -7
4 3 2 1 0 -1 -2 -3 -4 -5
6 5 4 3 2 1 0 -1 -2 -3
8 7 6 5 4 3 2 1 0 -1
10 9 8 7 6 5 4 3 2 1
12 11 10 9 8 7 6 5 4 3
14 13 12 11 10 9 8 7 6 5
16 15 14 13 12 11 10 9 8 7
18 17 16 15 14 13 12 11 10 9

Matriz C:
570 525 480 435 390 345 300 255 210 165
750 685 620 555 498 425 360 295 230 165
930 845 760 675 598 505 420 335 250 165
1110 1005 900 795 698 585 480 375 270 165
1290 1165 1040 915 790 665 540 415 290 165
1470 1325 1180 1035 890 745 600 455 310 165
1650 1485 1320 1155 990 825 660 495 330 165
1830 1645 1460 1275 1090 905 720 535 350 165
2010 1885 1600 1395 1190 985 780 575 370 165
2190 1965 1740 1515 1290 1065 840 615 390 165
```

Podemos ver en el código que N como tal es una variable estática, así que para editarlo usamos nano y cambiamos la variable para el caso de 1500, esto en cada una de las máquinas virtuales donde se va a ejecutar el programa.

The screenshot shows a Microsoft Word document titled "Reporte1GB - Word (Error de activación de productos)". The ribbon menu is visible at the top. A terminal window titled "Ubuntu@Prueba: ~" is embedded in the document, displaying Java code for matrix multiplication. The code uses long arrays for matrices A, B, and C, and temporary arrays A1 through C4. The terminal window has a title bar "GNU nano 2.9.3" and a status bar indicating "Modified". Below the terminal window, the Microsoft Word status bar shows various keyboard shortcuts for navigation and editing.

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.BufferedOutputStream;

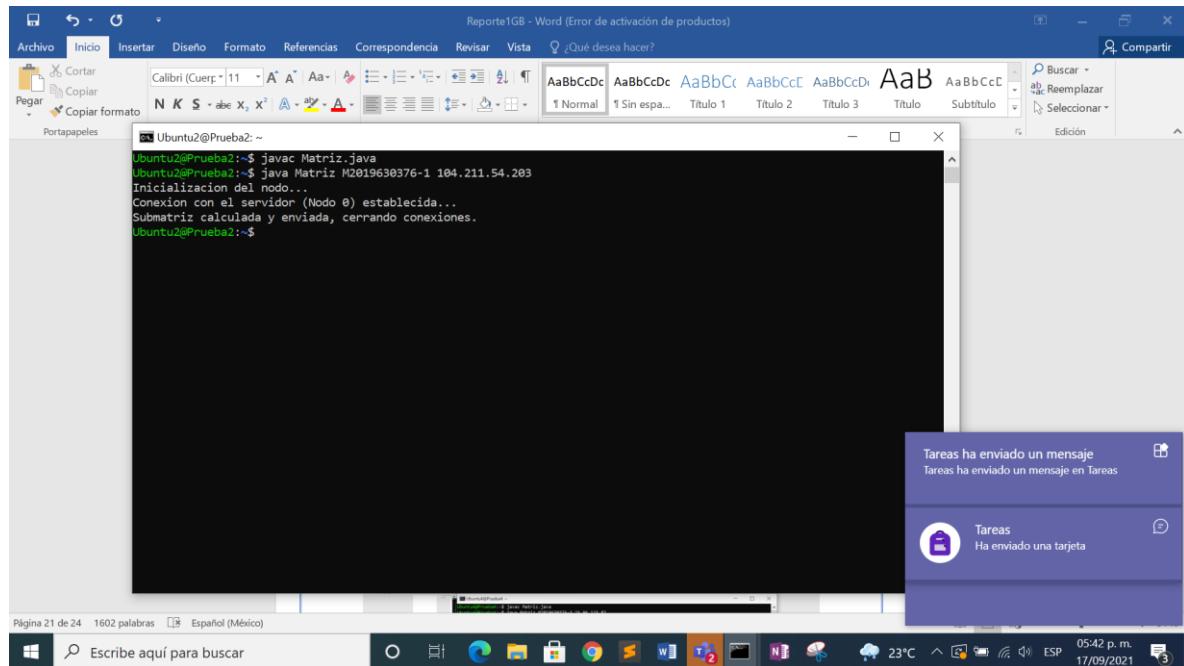
class Matriz
{
    static String ip;
    static String nodo;
    static int N = 1500;
    static long [][] A = new long[N][N];
    static long [][] B = new long[N][N];
    static long [][] C = new long[N][N];
    static long [][] A1 = new long[N/2][N];
    static long [][] B1 = new long[N/2][N];
    static long [][] A2 = new long[N/2][N];
    static long [][] B2 = new long[N/2][N];
    static long [][] C1;
    static long [][] C2;
    static long [][] C3;
    static long [][] C4;
```

Resultado de la ejecución del programa para N = 1500

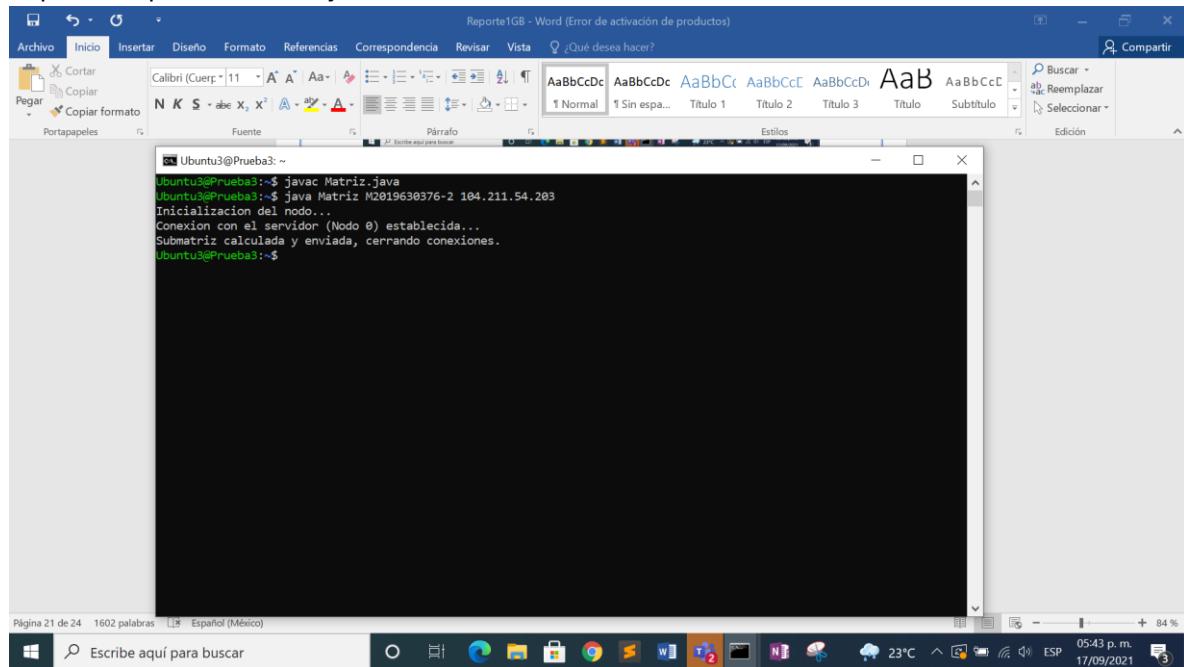
En esta captura podemos ver como compilamos nuevamente el programa y lo volvemos a ejecutar pasándole los mismos parámetros.

Ademas, podemos ver la checksum tal cual de la matriz resultado C, esta checksum es de **6953345718750000**.

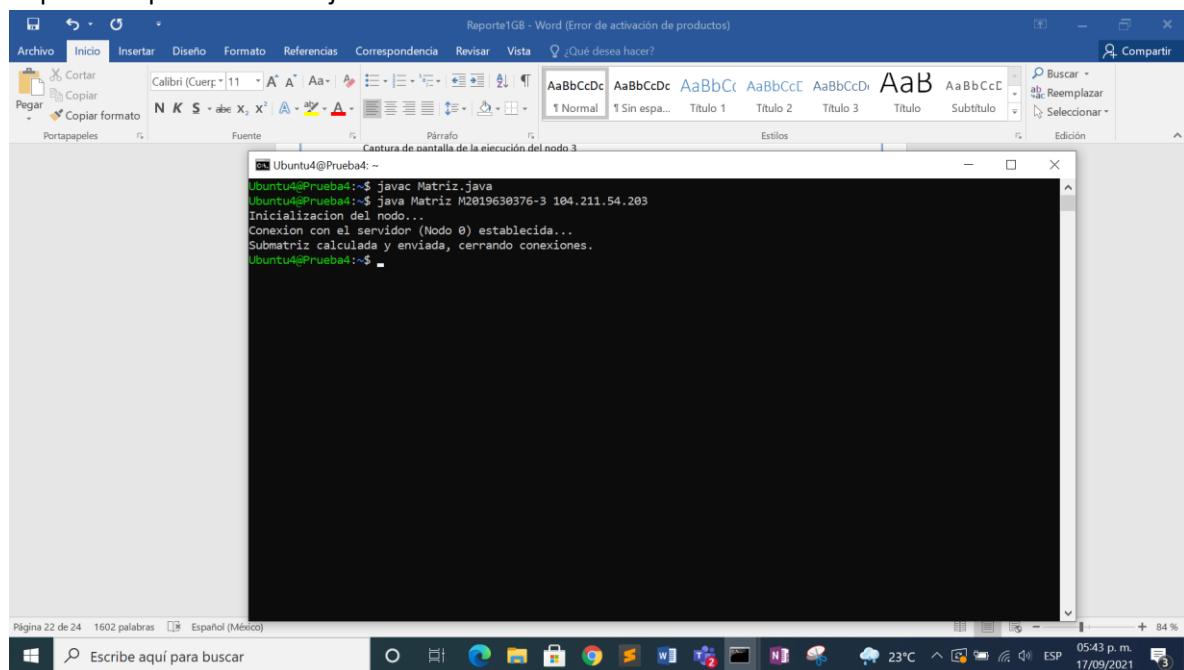
## Captura de pantalla de la ejecución del nodo 1



## Captura de pantalla de la ejecución del nodo 2



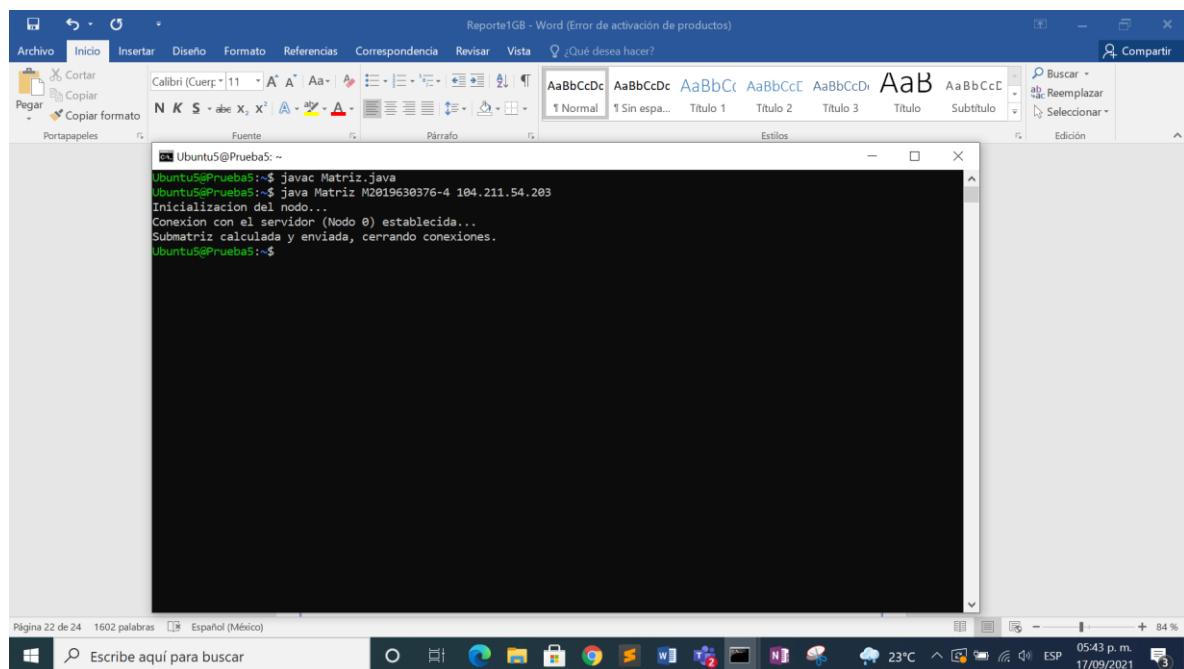
### Captura de pantalla de la ejecución del nodo 3



The screenshot shows a Microsoft Word document titled "Reporte1GB - Word (Error de activación de productos)". A terminal window is embedded within the document, displaying the following command-line output:

```
Ubuntu4@Prueba4:~$ javac Matriz.java
Ubuntu4@Prueba4:~$ java Matriz M2019630376-3 104.211.54.203
Inicialización del nodo...
Conexión con el servidor (Nodo 0) establecida...
Submatriz calculada y enviada, cerrando conexiones.
Ubuntu4@Prueba4:~$
```

### Captura de pantalla de la ejecución del nodo 4



The screenshot shows a Microsoft Word document titled "Reporte1GB - Word (Error de activación de productos)". A terminal window is embedded within the document, displaying the following command-line output:

```
Ubuntu5@Prueba5:~$ javac Matriz.java
Ubuntu5@Prueba5:~$ java Matriz M2019630376-4 104.211.54.203
Inicialización del nodo...
Conexión con el servidor (Nodo 0) establecida...
Submatriz calculada y enviada, cerrando conexiones.
Ubuntu5@Prueba5:~$
```

Finalmente, podemos ver las 5 máquinas virtuales creadas justo antes de que sean eliminadas para no agotar el crédito proporcionado por Azure.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes tabs for WhatsApp, SISDIS-4CV12: Tare..., (28) lofi hip hop rad..., Máquinas virtuales, Alumno del 3CV11, and CMP.ipynb - Colab... A user profile for 'doaxacap1500@alumno...' from 'INSTITUTO POLITÉCNICO NACIONAL' is visible. Below the header, the main content area is titled 'Máquinas virtuales' with a sub-header 'Instituto Politécnico Nacional (correo.ipn.mx)'. The page displays a table of 5 virtual machines, each with a checkbox, name, subscription, resource group, location, state, operating system, size, and IP address. The table is filtered by 'Suscripción == todo', 'Grupo de recursos == todo', and 'Ubicación == todo'. At the bottom, there are navigation links for '< Anterior', 'Página 1 de 1', and 'Siguiente >'. The taskbar at the bottom of the screen shows various application icons and the date/time '17/09/2021 05:43 p.m.'

| Nombre  | Suscripción            | Grupo de recursos | Ubicación       | Estado       | Sistema operativo | Tamaño          | Dirección IP públ.. |
|---------|------------------------|-------------------|-----------------|--------------|-------------------|-----------------|---------------------|
| Prueba  | Azure para estudiantes | grupo_pru...      | Este de EE. UU. | En ejecución | Linux             | Standard_B1s    | 104.211.54.203      |
| Prueba2 | Azure para estudiantes | grupo_pru...      | Este de EE. UU. | En ejecución | Linux             | Standard_B1s    | 20.85.227.49        |
| Prueba3 | Azure para estudiantes | grupo_pru...      | Este de EE. UU. | En ejecución | Linux             | Standard_B1s    | 52.136.116.153      |
| Prueba4 | Azure para estudiantes | grupo_pru...      | Este de EE. UU. | En ejecución | Linux             | Standard_B1s    | 20.85.224.37        |
| Prueba5 | Azure para estudiantes | grupo_pru...      | Este de EE. UU. | En ejecución | Linux             | Standard_DS1_v2 | 40.121.52.158       |

## Conclusiones

Esta tarea fue bastante interesante de realizar y ciertamente, más retadora que las anteriores, se puso mucho de lo aprendido hasta la fecha en clase para su realización, además de que pudimos observar como genuinamente puede trabajar un programa de manera distribuida, haciendo un cálculo que previamente ya habíamos hecho mucho más rápido aun, además de que aplicamos muchas de las actividades vistas en clase para la creación de las máquinas virtuales además de aprender algunos conceptos adicionales para llevar a cabo la tarea. En general creo que fue de mucha ayuda para repasar algunos de los conceptos vistos en clase previamente.

## Anexos

### Código

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.BufferedOutputStream;

/*
 * @author: David Oaxaca
 *
 * Forma de compilarlo:
 * javac Matriz.java
 *
 * Forma de ejecutarlo:
 * java Matriz M2019630376-0 104.211.54.203
 *
 */
*****
```

```
class Matriz
{
    static String ip;
    static String nodo;
    static int N = 1500;
    static long [][] A = new long[N][N];
    static long [][] B = new long[N][N];
    static long [][] C = new long[N][N];
    static long [][] A1 = new long[N/2][N];
    static long [][] B1 = new long[N/2][N];
    static long [][] A2 = new long[N/2][N];
    static long [][] B2 = new long[N/2][N];
    static long [] C1;
    static long [] C2;
    static long [] C3;
    static long [] C4;
```

```
static class Worker extends Thread{
```

```
    Socket conexion;
    Worker(Socket conexion){
        this.conexion = conexion;
```

```

}

public void run(){
try{

    System.out.println("Nodo conectado, iniciando calculo de multiplicacion de
matrices...");

    //Declaramos object streams para enviar las submatrices serializadas como un objeto
(Y tambien un String)

    ObjectOutputStream object_salida = new
ObjectOutputStream(conexion.getOutputStream());
    ObjectInputStream object_entrada = new
ObjectInputStream(conexion.getInputStream());

    //Los objetos recibidos los casteamos a una variable, en este caso, de tipo String
String id_nodo = (String) object_entrada.readObject();

    //De acuerdo con el ID de nodo recibido enviara las submatrices correspondientes

    switch(id_nodo){

        case "M2019630376-1":
            object_salida.writeObject(A1);
            object_salida.writeObject(B1);

            C1 = (long[][]) object_entrada.readObject();

            //Impresión de la submatriz que esta recibiendo
            //System.out.println("Matriz C1 recibida: ");
            //matrixPrint(C1);
            break;

        case "M2019630376-2":
            object_salida.writeObject(A1);
            object_salida.writeObject(B2);

            C2 = (long[][]) object_entrada.readObject();

            //Impresión de la submatriz que esta recibiendo
            //System.out.println("Matriz C2 recibida: ");
            //matrixPrint(C2);
            break;

        case "M2019630376-3":
            object_salida.writeObject(A2);
    }
}

```

```

object_salida.writeObject(B1);

C3 = (long[][])) object_entrada.readObject();

//Impresión de la submatriz que esta recibiendo
//System.out.println("Matriz C3 recibida: ");
//matrixPrint(C3);
break;
case "M2019630376-4":
    object_salida.writeObject(A2);
    object_salida.writeObject(B2);

    C4 = (long[][])) object_entrada.readObject();

//Impresión de la submatriz que esta recibiendo
//System.out.println("Matriz C4 recibida: ");
//matrixPrint(C4);
break;
default:
    System.out.println("No es un nodo registrado para este sistema");
    break;

}

object_entrada.close();
object_salida.close();
conexion.close();

} catch(Exception e){
    e.printStackTrace();
}

}

}

public static void main(String[] args) throws Exception
{
//Validamos el recibimiento de los parametros necesarios para la topologia
if(args.length != 2){

    System.err.println("Se debe pasar como parametros el numero del nodo y la IP del
servidor (En caso de ser el nodo 0 pasar su propia dirección IP)");
    System.exit(1);

}

```

```

nodo = args[0];
ip = args[1];

if(nodo.equals("M2019630376-0")){
    System.out.println("Servidor (Nodo 0) iniciado...");

    // inicializa las matrices A y B

    for (int i = 0; i < N; i++)
        for (int j = 0; j < N; j++)
    {
        A[i][j] = 2 * i + j;
        B[i][j] = 2 * i - j;
        C[i][j] = 0;
    }

    // Declaración de la matriz donde guardaremos a B transpuesta

    long [][] B_transpuesta = new long[N][N];

    // Copiamos la matriz B a la matriz donde faremos su transpuesta

    for (int i = 0; i < N; i++)
        for (int j = 0; j < N; j++)
    {
        B_transpuesta[i][j] = B[i][j];
    }

    // transpone la matriz B, guardada en la matriz B_transpuesta

    for (int i = 0; i < N; i++)
        for (int j = 0; j < i; j++)
    {
        long x = B_transpuesta[i][j];
        B_transpuesta[i][j] = B_transpuesta[j][i];
        B_transpuesta[j][i] = x;
    }

    // Creación de A1 y A2

    for (int i = 0; i < N/2; i++) {
        for (int j = 0; j < N; j++) {
            A1[i][j] = A[i][j];
        }
    }
}

```

```

        A2[i][j] = A[i + N/2][j];
    }
}

// Creación de B1 y B2

for (int i = 0; i < N/2; i++) {
    for (int j = 0; j < N; j++) {
        B1[i][j] = B_transpuesta[i][j];
        B2[i][j] = B_transpuesta[i + N/2][j];
    }
}

// Inicializamos un server socket en el puerto 20000

ServerSocket servidor = new ServerSocket(20000);

// Creamos 4 objetos Worker que seran los hilos que enviaran las matrices a los otros nodos

Worker[] hilos = new Worker[4];

// Esperamos las conexiones propiamente para inicializar los hilos pasando la conexión

for(int i = 0; i < 4; i++){
    Socket conexion = servidor.accept();
    hilos[i] = new Worker(conexion);
    hilos[i].start();
}

for(int i = 0; i < 4; i++){
    hilos[i].join();
}

// Une las matrices C1, C2, C3 y C4 para formar la matriz C, producto de las matrices A y B

for (int i = 0; i < N/2; i++) {
    for (int j = 0; j < N/2; j++) {
        C[i][j] = C1[i][j];
        C[i][j + N/2] = C2[i][j];
        C[i + N/2][j] = C3[i][j];
        C[i + N/2][j + N/2] = C4[i][j];
    }
}

```

```

//Calculo y despliegue del checksum de la matriz C
System.out.println("\nChecksum de la matriz C: " + checksum(C) + "\n");

//Si N = 10, desplegamos las matrices A, B (Sin estar traspuesta) y C
if(N == 10){

    System.out.println("Matriz A: \n");
    matrixPrint(A);

    System.out.println("Matriz B: \n");
    matrixPrint(B);

    System.out.println("Matriz C: \n");
    matrixPrint(C);

}

}else{

    System.out.println("Inicializacion del nodo...");

    //Inicializamos un socket con lo que es la dirección del servidor pasada como
    parametro y el puerto 20000

    Socket conexion = new Socket(ip, 20000);

//Declaramos object streams para enviar las submatrices serializadas como un objeto

    ObjectOutputStream object_salida = new
ObjectOutputStream(conexion.getOutputStream());
    ObjectInputStream object_entrada = new
ObjectInputStream(conexion.getInputStream());

    System.out.println("Conexion con el servidor (Nodo 0) establecida...");

    object_salida.writeObject(nodo);
    object_salida.flush();

//Los objetos recibidos los casteamos a una variable, en este caso, de tipo long[][]

    long [][] Ax = (long[][])object_entrada.readObject();

//Impresión de la submatriz que esta recibiendo
//System.out.println("Matriz A recibida: ");

```

```

//matrixPrint(Ax);

long [][] Bx = (long[][])(object_entrada.readObject());

//Impresión de la submatriz que esta recibiendo
//System.out.println("Matriz B recibida: ");
//matrixPrint(Bx);

object_salida.writeObject(multiplicaMatriz(Ax, Bx, true));

System.out.println("Submatriz calculada y enviada, cerrando conexiones.");

object_salida.close();
object_entrada.close();
conexion.close();

}

}

/*
*
* * Parametros: Dos matrices de tipo long (long [][] ) las cuales multiplicara y un boolean para saber si la matriz 2 esta transpuesta
* * Return: Una matriz de tipo long (long[][]) la cual sera el producto de las dos matrices pasadas
* * Brief: Esta funcion obtiene la matriz producto de dos matrices y dependiendo si es transpuesta sus dimensiones (mxn) en caso de que no sean del mismo tamaño
*/

```

```

public static long [][] multiplicaMatriz(long [][] matrix1, long [][] matrix2, boolean transpuesta){

    int cols = matrix1.length;

    //Si la matriz 2 (matrix2) esta transpuesta usa el tamaño de las columnas para crear el tamaño de las filas de la nueva matriz
    int rows = transpuesta ? matrix2.length : matrix2[0].length;

    long [][] res = new long[cols][rows];

    for (int i = 0; i < cols; i++)
        for (int j = 0; j < rows; j++)

```

```

for (int k = 0; k < N; k++)
    res[i][j] += matrix1[i][k] * matrix2[j][k];

return res;

}

/*
*
* Parametros: Una matriz de tipo long (long [][]) la cual se imprimira con un formato
* Return: No retorna nada
* Brief: Imprime una matriz pasada como parametro usando printf() para darle un formato
*
*/

```

```

public static void matrixPrint(long [][] matrix){

    for (int i = 0; i < matrix.length; i++){
        for (int j = 0; j < matrix[i].length; j++)
        {

            //Para imprimir utilizamos un printf, que al igual que en C o Python, nos permite
            darle un formato a la cadena
            //En este caso, dando a entender que pasaremos 5 "argumentos", uno de ellos sera el
            numero, el resto se usaran como espacios

            System.out.printf("%5d", matrix[i][j]);

        }

        System.out.println("");


    }

    System.out.println("\n");

}

/*
*
* Parametros: Una matriz de tipo long (long [][]) a la cual se le calculara el checksum
* Return: Una variable de tipo long contenido el checksum de la matriz pasada

```

*\* Brief: Esta función calcula la checksum de una matriz pasada, mediante la suma de todo sus elementos*

```

*
*/

```

```

public static long checksum(long [][] matrix){

    long checksum = 0;

    for (int i = 0; i < matrix.length; i++){
        for (int j = 0; j < matrix[i].length; j++){

            checksum += matrix[i][j];
        }
    }

    return checksum;
}
}

```

## Creación de la quinta máquina virtual (Cambio a 3.5 GB de RAM)

Captura de pantalla que muestra el cambio de familia para la quinta máquina virtual.

