



INSTITUTO POLITÉCNICO NACIONAL



ESCUELA SUPERIOR DE CÓMPUTO

Desarrollo de Sistemas Distribuidos

**“Tarea 5: Multiplicación de matrices utilizando
objetos distribuidos”**

Grupo: 4CV13

Alumno:

Godinez Morales Mario Sebastian

Profesor:

Pineda Guerrero Carlos

Desarrollo

Creación de la máquina virtual

En la página principal de azure seleccionamos máquinas virtuales, posteriormente en el apartado de VM seleccionamos máquinas virtuales y crear como se muestra en la imagen 1.

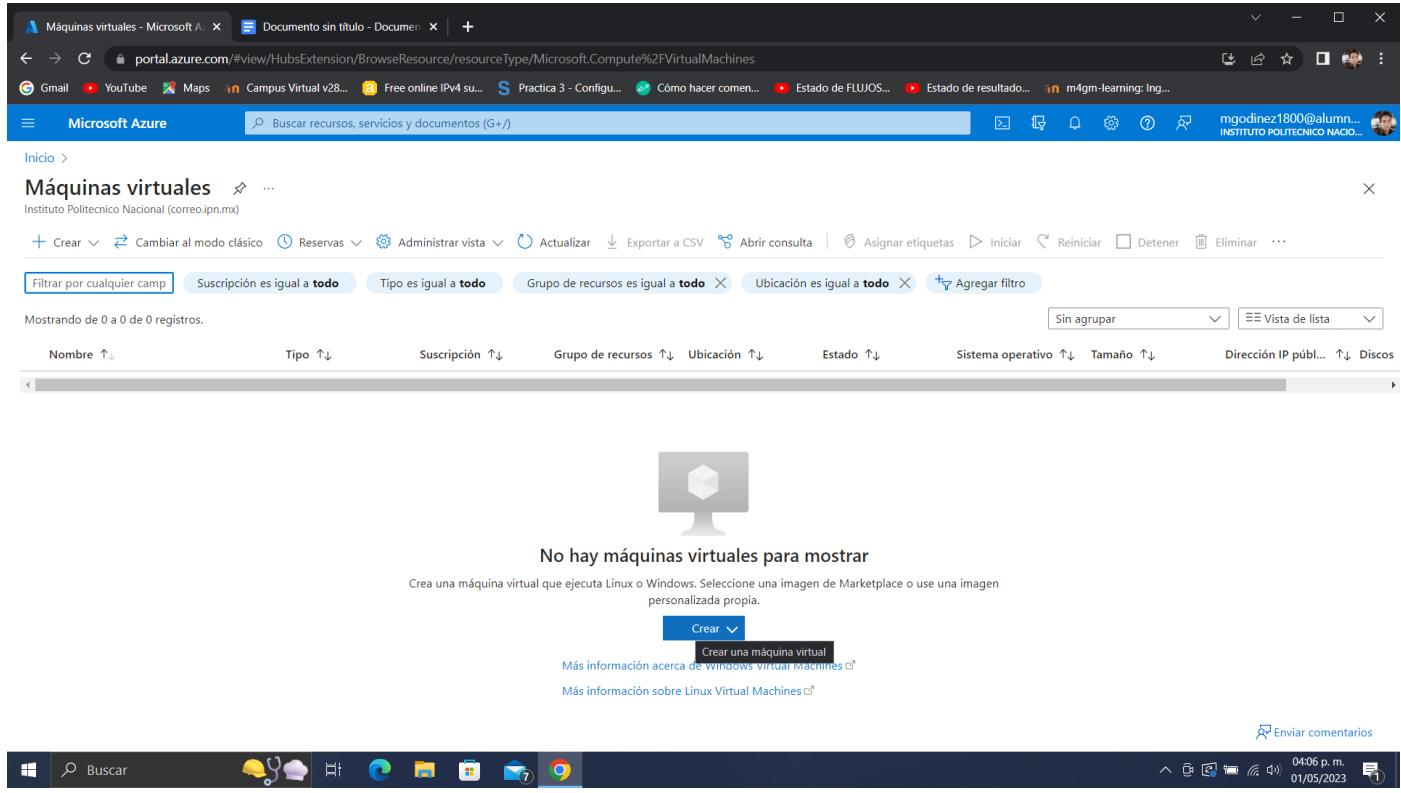


Imagen 1

Posteriormente se configura la información básica de la VM como el grupo de recursos, la región de la VM, el nombre para nuestro caso se llama T5-2019630034-0 para el nodo 0 (cliente RMI) como se muestra en la imagen 2.

The screenshot shows the 'Create a virtual machine' wizard in Microsoft Azure. The current step is 'Datos básicos' (Basic details). The page title is 'Crear una máquina virtual' (Create a virtual machine). The top navigation bar includes tabs for 'Discos', 'Redes', 'Administración', 'Monitoring', 'Opciones avanzadas', 'Etiquetas', and 'Revisar y crear'. A note at the top states: 'Cree una máquina virtual que ejecuta Linux o Windows. Seleccione una imagen de Azure Marketplace o use una imagen personalizada propia. Complete la pestaña Conceptos básicos y, después, use Revisar y crear para aprovisionar una máquina virtual con parámetros predeterminados o bien revise cada una de las pestañas para personalizar la configuración.' Below this is a warning message: 'Es posible que esta suscripción no sea apta para implementar máquinas virtuales de ciertos tamaños en determinadas regiones.' The 'Suscripción' dropdown is set to 'Azure for Students'. The 'Grupo de recursos' dropdown is set to '(Nuevo) Tarea5'. The 'Nombre de máquina virtual' input field contains 'T5-2019630034-0'. The 'Región' dropdown is set to '(US) East US'. At the bottom, there are buttons for 'Revisar y crear' (Review + Create), '< Anterior' (Previous), 'Siguiente: Discos >' (Next: Disks), and 'Enviar comentarios' (Send feedback). The status bar at the bottom right shows the date and time: '01/05/2023 04:13 p.m.'

Imagen 2

Configuramos el hardware de la VM, la contraseña de acceso y los puertos de entrada como se muestra en la imagen 3.

The screenshot shows the Microsoft Azure portal interface for creating a new virtual machine. The page title is "Crear una máquina virtual".

Imagen *: Ubuntu Server 18.04 LTS - x64 Gen2

Arquitectura de VM: x64

Ejecución de Azure Spot con descuento: Desactivado

Tamaño *: Standard_B1s - 1 vCPU, 1 GiB de memoria (7,59 US\$/mes)

Cuenta de administrador

Tipo de autenticación: Contraseña

Nombre de usuario *: mario

Contraseña *: (obscuro)

Confirmar contraseña *: (obscuro)

Reglas de puerto de entrada: Sección desactivada.

Botones de navegación: "Revisar y crear" (destacado), "< Anterior", "Siguiente: Discos >".

Barra superior: Buscar recursos, servicios y documentos (G+).

Barra inferior: Estado de conexión, fecha y hora (04:24 p. m., 01/05/2023).

Imagen 3

Damos click en siguiente y posteriormente configuraremos los discos, en esta parte seleccionamos el tamaño y tipo de almacenamiento de 30GB con un HDD estándar como se muestra en la imagen 4.

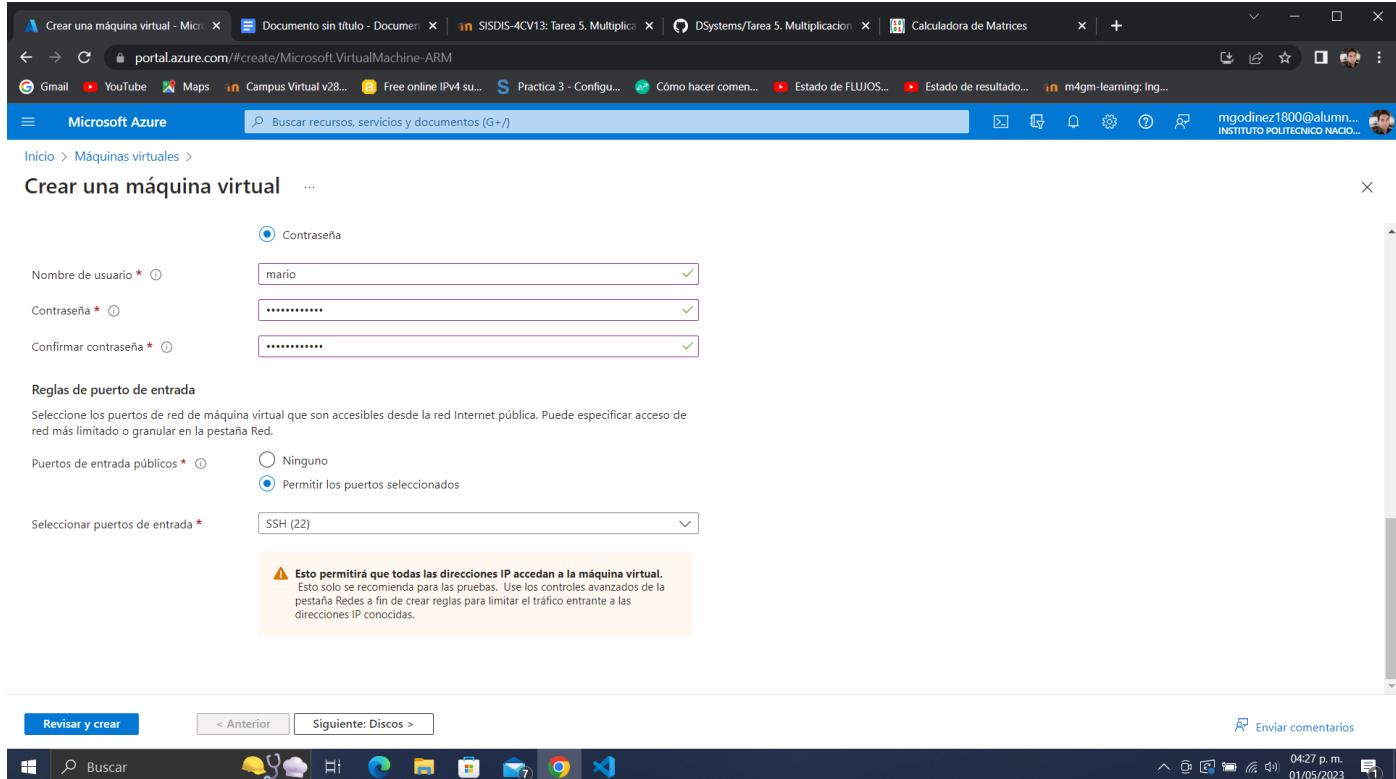


Imagen 4

En el apartado de redes no configuramos nada

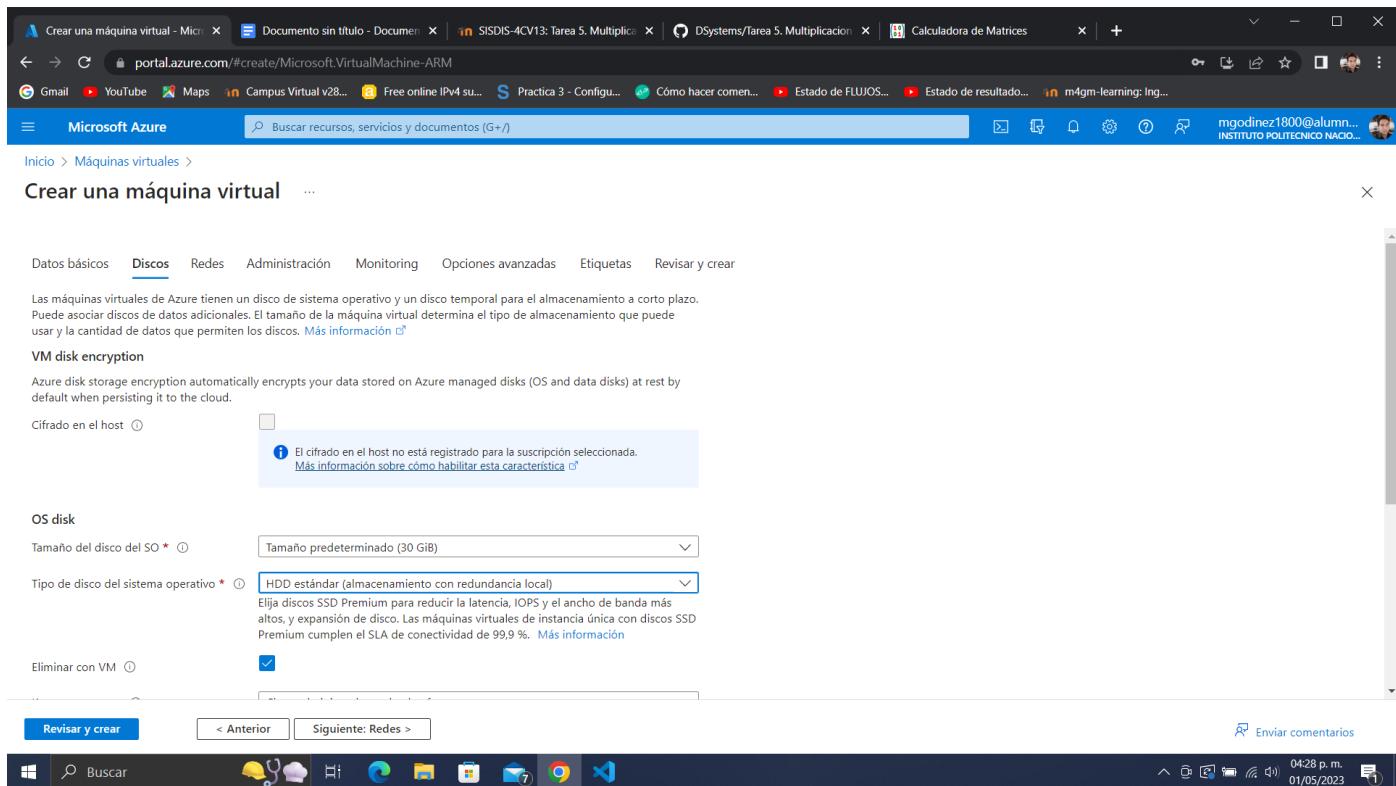


Imagen 5

En la parte de administración no configuramos nada.

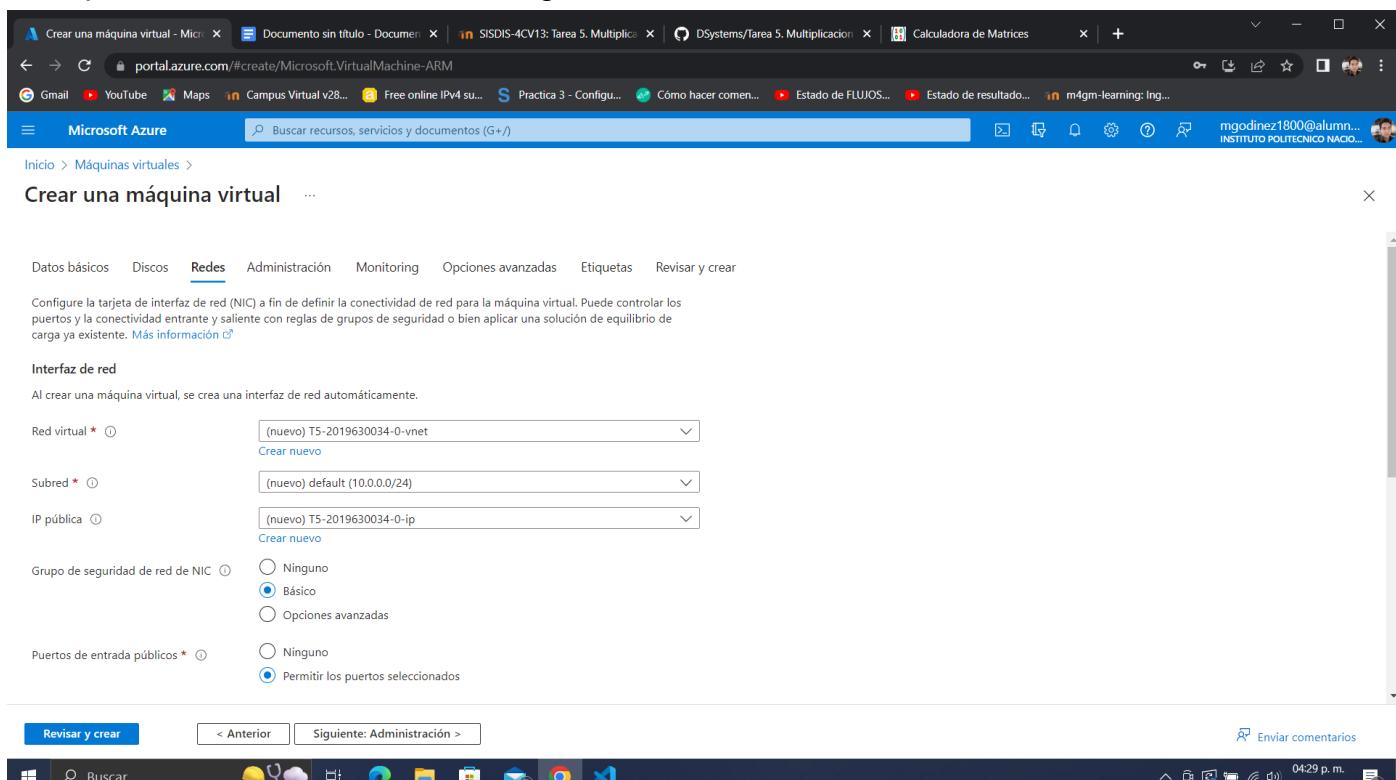


Imagen 6

Damos click en revisar y crear como se muestra en la imagen 7.

The screenshot shows the Microsoft Azure portal interface for creating a new virtual machine. The top navigation bar includes links like 'Crear una máquina virtual - Micr...', 'Documento sin título - Documento...', 'SISDIS-4CV13: Tarea 5. Multiplic...', 'DSystems/Tarea 5. Multiplicacion...', 'Calculadora de Matrices...', 'portal.azure.com/#create/Microsoft.VirtualMachine-ARM', 'Gmail', 'YouTube', 'Maps', 'Campus Virtual v28...', 'Free online IPv4 su...', 'Práctica 3 - Configu...', 'Cómo hacer coment...', 'Estado de FLUJOS...', 'Estado de resultado...', 'm4gm-learning: Ing...', and 'mgodinez1800@alumn... INSTITUTO POLITÉCNICO NACIONAL'. The main title is 'Crear una máquina virtual'. Below it, tabs include 'Datos básicos', 'Discos', 'Redes', 'Administración' (which is selected), 'Monitoring', 'Opciones avanzadas', 'Etiquetas', and 'Revisar y crear'. A sub-header says 'Configure las opciones de supervisión y administración de la VM.' Under 'Administración', there's a section for 'Microsoft Defender for Cloud' with a note about统一管理 security features across the cloud. There are checkboxes for enabling basic plan and system-assigned managed identities. The 'Azure AD' section includes a note about using certificate-based authentication for SSH. At the bottom, there are 'Revisar y crear' and 'Siguiente: Monitoring >' buttons, along with a 'Enviar comentarios' link. The status bar at the bottom right shows the date and time: '04:30 p.m. 01/05/2023'.

Imagen 7

Revisamos si la información es correcta y seleccionamos crear como se muestra en la imagen 8.

The screenshot shows the Microsoft Azure portal interface for creating a new virtual machine, specifically on the 'Create' step. The top navigation bar is identical to the previous screenshot. The main title is 'Crear una máquina virtual'. Below it, tabs include 'Datos básicos', 'Discos', 'Redes', 'Administración' (selected), 'Monitoring', 'Opciones avanzadas', 'Etiquetas', and 'Revisar y crear' (selected). A green success message 'Validación superada' is displayed. A note about estimated costs is present. The 'Price' section shows '1 X Standard B1s by Microsoft' with a price of '0,0104 USD/hr' and a link to 'Pricing for other VM sizes'. The 'TERMS' section contains legal text about agreeing to terms and privacy statements. At the bottom, there are 'Crear' and 'Siguiente: Descargar una plantilla para la automatización >' buttons, along with a 'Enviar comentarios' link. The status bar at the bottom right shows the date and time: '04:31 p.m. 01/05/2023'.

Imagen 8

Esperamos a que la implementación termine como se muestra en la imagen 9

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes tabs for 'CreateVm-CanonicalUbuntuServer' (active), 'Documento sin título - Documento', 'SISDIS-4CV13: Tarea 5. Multiplicacion', 'DSystems/Tarea 5. Multiplicacion', and 'Calculadora de Matrices'. Below the navigation bar, the address bar displays 'portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fsubscriptions%2F057dbe04-e860-4a02-b4e2-eee84f97b246%2FresourceGroups%2FTarea5%2Fprovider...'.

The main content area shows a deployment named 'CreateVm-CanonicalUbuntuServer-18_04-Its-gen2-20230501160629'. The status is 'La implementación está en curso' (The implementation is in progress). It provides details: Nombre de implementación: CreateVm-CanonicalUbuntuServer-18...; Hora de inicio: 1/5/2023, 16:31:47; Suscripción: Azure for Students; Grupo de recursos: Tarea5. A 'Actualizar' button is available.

A sidebar on the right contains links for 'Microsoft Defender for Cloud', 'Tutoriales gratuitos de Microsoft', 'Trabajar con un experto', and 'Buscar un experto de Azure'.

The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date and time as 01/05/2023 at 04:32 p.m.

Imagen 9

Una vez implementado el recurso seleccionamos ir al recurso como se muestra en la imagen 10.

The screenshot shows the Microsoft Azure portal with the URL <https://portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/>. The main content area displays the deployment status for 'CreateVm-Canonical.UbuntuServer-18_04-Its-gen2-20230501160629'. A prominent green checkmark indicates that the implementation has completed successfully. Below this, there are sections for 'Detalles de implementación' (Implementation details) and 'Pasos siguientes' (Next steps), which include recommended actions like 'Configurar el apagado automático' (Configure automatic shutdown). On the right side of the portal, there are several service cards: 'Cost Management' (with a note about staying within budget), 'Microsoft Defender for Cloud' (protecting applications and infrastructure), and 'Tutoriales gratuitos de Microsoft' (free Microsoft tutorials). The bottom of the screen shows the Windows taskbar with various pinned icons.

Imagen 10

Y la VM estará creada como se muestra en la imagen 11.

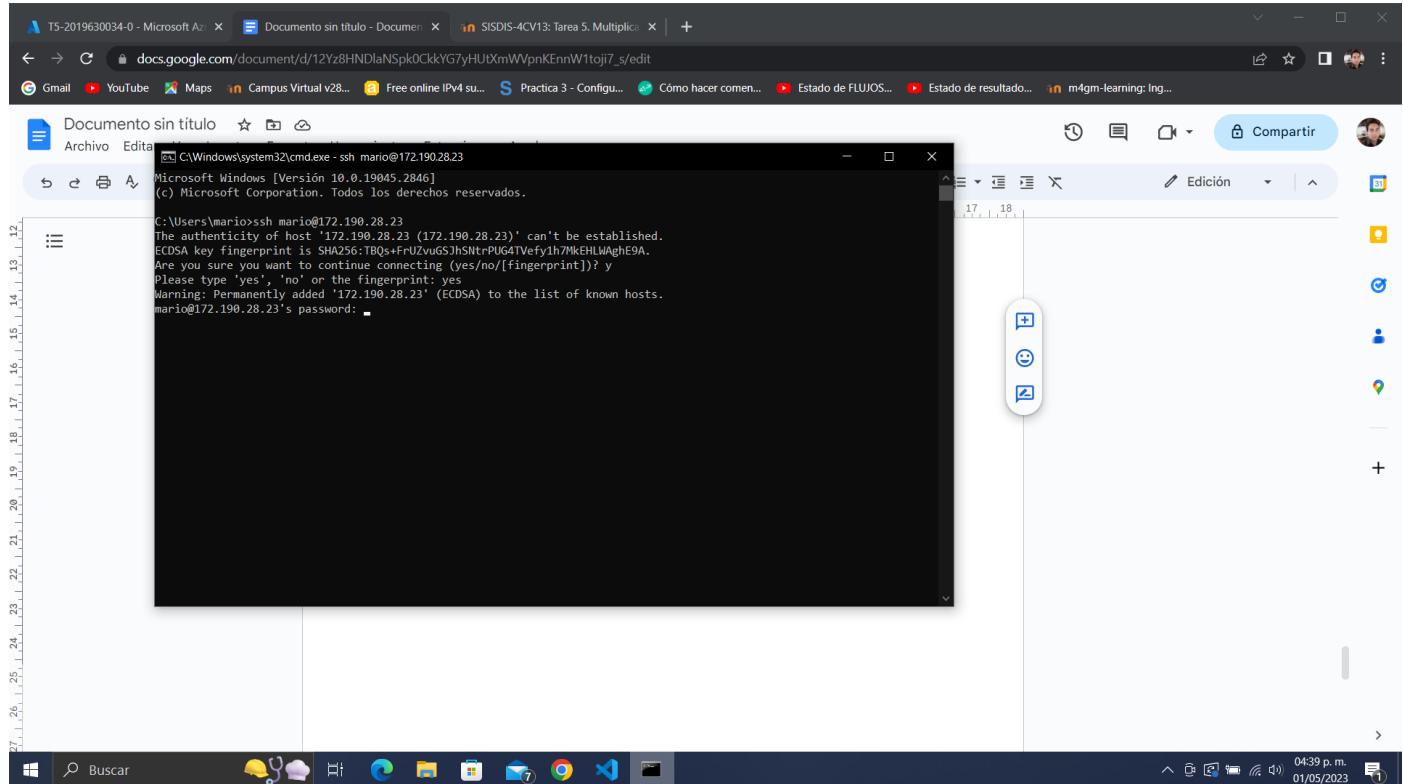
The screenshot shows the Microsoft Azure portal with the URL <https://portal.azure.com/#/resource/subscriptions/057dbe04-e860-4a02-b4e2-eee84f97b246/resourcegroups/Tarea5/providers/Microsoft.Compute/virtualMachines/T5-2019630034-0>. The main content area displays the details for the virtual machine 'T5-2019630034-0'. Key information shown includes the operating system (Linux (ubuntu 18.04)), location (East US), and network settings (IP addresses: 172.190.28.23 and 172.190.28.24). The left sidebar shows the navigation menu with 'Información general' selected. The bottom of the screen shows the Windows taskbar with various pinned icons.

Imagen 11

Posteriormente hacemos los mismos pasos para la creación de las 2 VM restantes.

Conexión a la VM

Para conectarnos a la VM que se ha creado se utiliza el comando ssh usuario@ip pública como se muestra en la imagen 12.



The screenshot shows a Microsoft Edge browser window with several tabs open. The active tab is a Google Docs document titled "Documento sin título". Inside the document, there is a terminal window running on Windows 10. The command entered is "ssh mario@172.190.28.23". The terminal output shows the following:

```
C:\Users\mario>ssh mario@172.190.28.23
The authenticity of host '172.190.28.23 (172.190.28.23)' can't be established.
ECDSA key fingerprint is SHA256:TQo+FrUzVuG5HSNtPw64TVefyIh7MkEHlWAghE9A.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '172.190.28.23' (ECDSA) to the list of known hosts.
mario@172.190.28.23's password: 
```

Imagen 12

Introducimos la contraseña y si todo es correcto nos conectará a la VM como se muestra en la imagen 13.

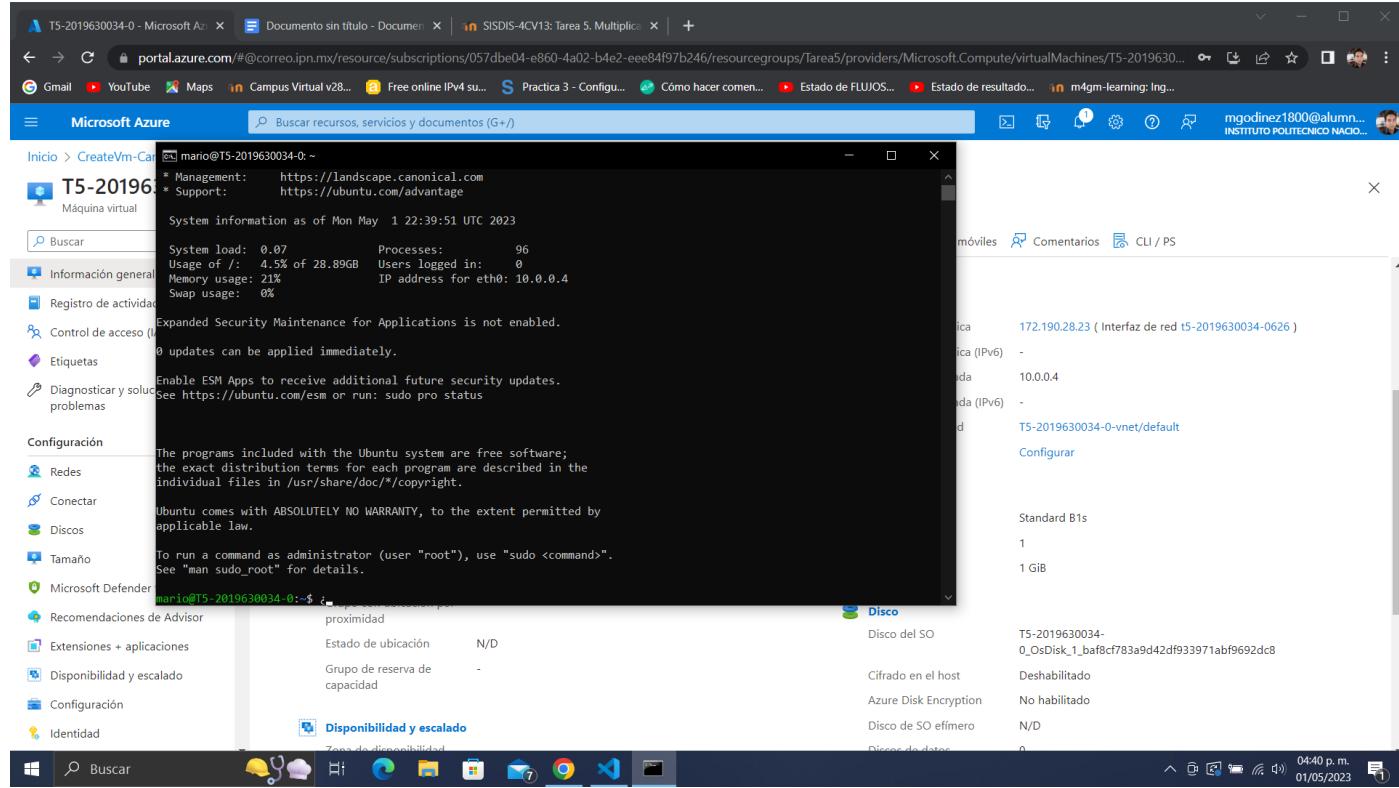
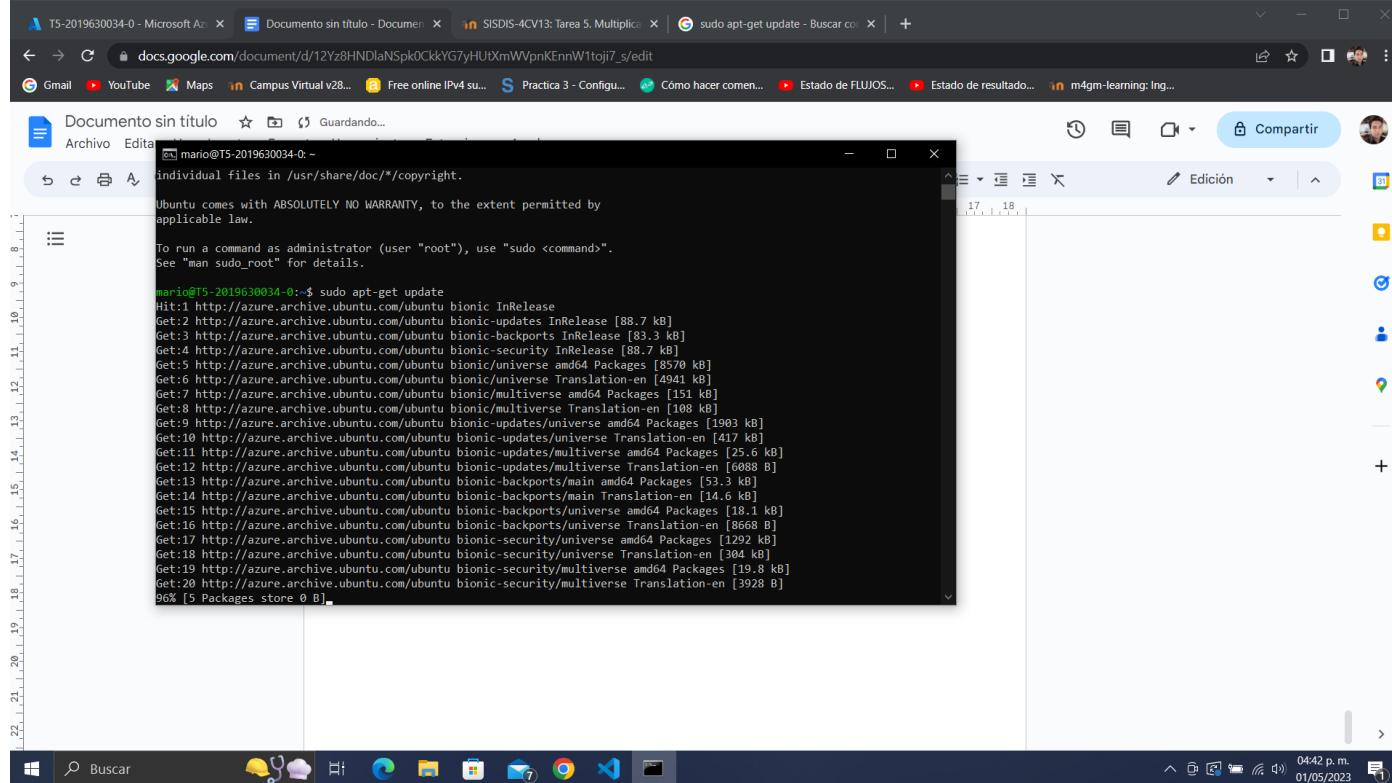


Imagen 13

Instalación de Java en la VM

Para instalar java en la VM una vez se tiene conexión a la VM se ejecuta el comando sudo apt-get update para actualizar los paquetes como se muestra en la imagen 14.



```
mario@T5-2019630034-0: ~$ sudo apt-get update
Hit:1 http://azure.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu bionic/universe Translation-en [494 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:8 http://azure.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:9 http://azure.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1903 kB]
Get:10 http://azure.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [417 kB]
Get:11 http://azure.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [25.6 kB]
Get:12 http://azure.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [6088 B]
Get:13 http://azure.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [53.3 kB]
Get:14 http://azure.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [14.6 kB]
Get:15 http://azure.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [18.1 kB]
Get:16 http://azure.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [8668 B]
Get:17 http://azure.archive.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1292 kB]
Get:18 http://azure.archive.ubuntu.com/ubuntu bionic-security/universe Translation-en [304 kB]
Get:19 http://azure.archive.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [19.8 kB]
Get:20 http://azure.archive.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [3928 B]
96% [5 Packages store 0 B]
```

Imagen 14

Una vez actualizados ejecutamos el comando sudo apt install default-jre para instalar java y esperamos a que termine como se muestra en la imagen 15.

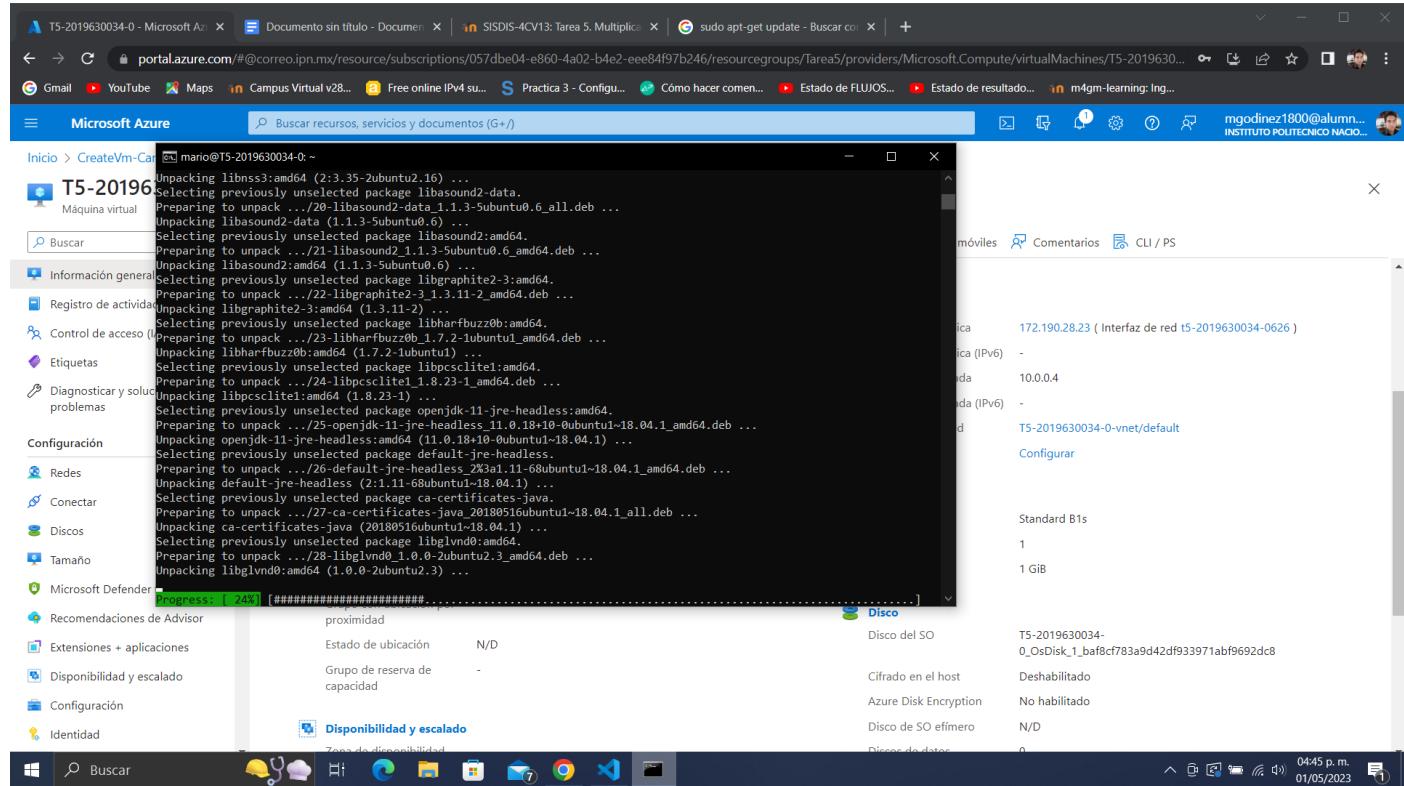
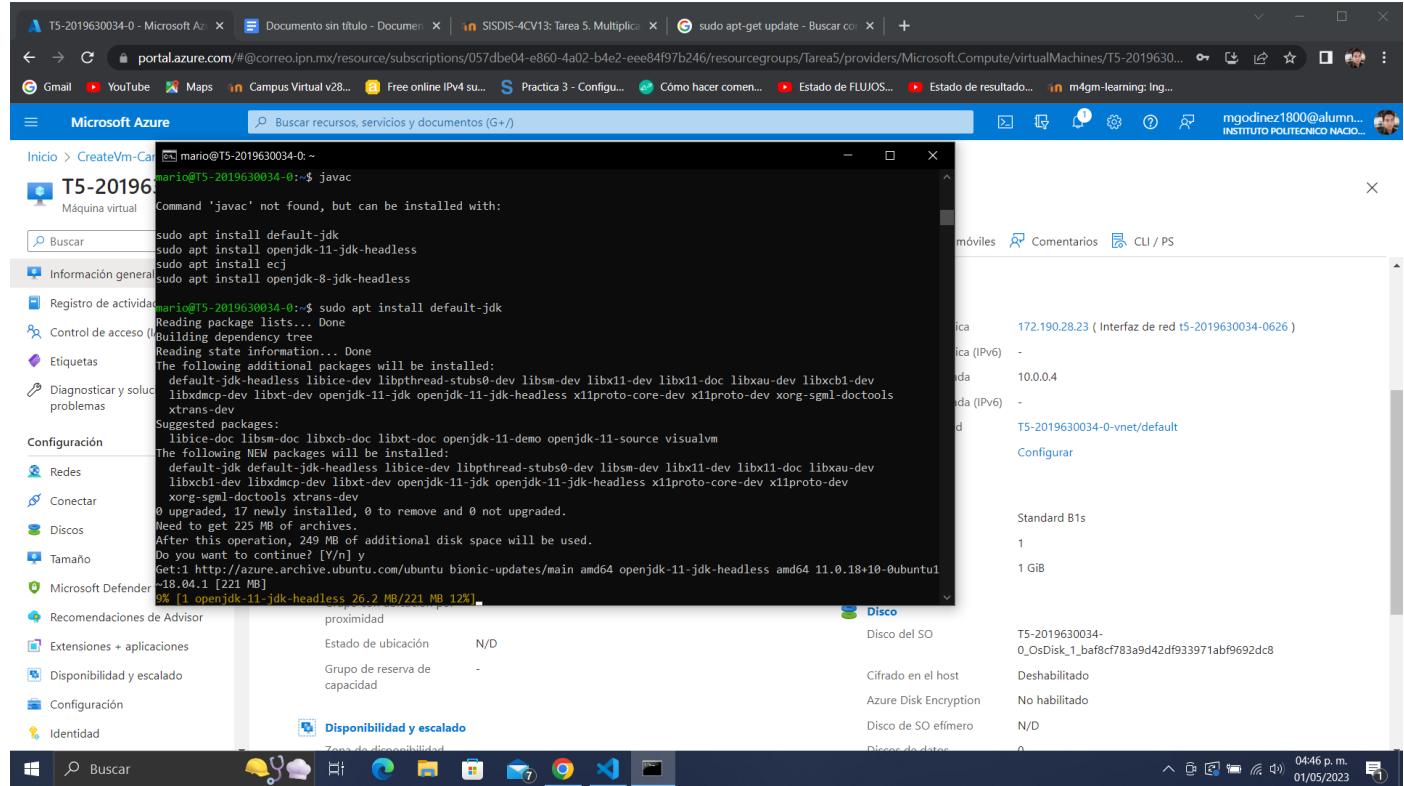


Imagen 15

Posteriormente instalamos el jdk con el comando sudo apt install openjdk-11-jre-headless para el rmiregistry y el default jdk con el comando sudo apt install default-jdk para javac y esperamos a que termine de instalarse como se muestra en la imagen 16.



The screenshot shows a Microsoft Azure VM terminal window titled 'T5-20196'. The user is running the command 'javac' which is not found, so they run 'sudo apt install default-jdk'. The terminal output shows the package being installed along with its dependencies. The right side of the screen displays the Azure VM configuration details, including network settings (IPv4 and IPv6), disk information (Standard B1s, 1 vCore, 1 GiB), and security features like Azure Disk Encryption.

```
mario@T5-20196:~$ javac
Command 'javac' not found, but can be installed with:
  sudo apt install default-jdk
  sudo apt install openjdk-11-jdk-headless
  sudo apt install ecj
  sudo apt install openjdk-8-jdk-headless
mario@T5-20196:~$ sudo apt install default-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  default-jdk-headless libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libx11-doc libxau-dev libxcb1-dev
  libxdmcp-dev libxt-dev openjdk-11-jdk openjdk-11-jdk-headless x11proto-core-dev x11proto-dev xorg-sgml-doctools
  xtrans-dev
Suggested packages:
  libice-doc libsm-doc libxcb-doc libxt-doc openjdk-11-demo openjdk-11-source visualvm
The following NEW packages will be installed:
  default-jdk default-jdk-headless libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libx11-doc libxau-dev
  libxcb1-dev libxdmcp-dev libxt-dev openjdk-11-jdk openjdk-11-jdk-headless x11proto-core-dev x11proto-dev
  xorg-sgml-doctools xtrans-dev
0 upgraded, 17 newly installed, 0 to remove and 0 not upgraded.
Need to get 225 MB of archives.
After this operation, 249 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 openjdk-11-jdk-headless amd64 11.0.18+10-0ubuntu1
[18.04.1/221 MB]
9% [openjdk-11-jdk-headless 26.2 MB/221 MB 12%]
Disco
```

Disco	Disco del SO
T5-20196:0_OsDisk_1_baf8cf783a9d42df933971abf9692dc8	Disco del SO
Cifrado en el host	Deshabilitado
Azure Disk Encryption	No habilitado
Disco de SO efímero	N/D
Disco de datos	0

Imagen 16

De esta manera ya tenemos instalado java en la VM, repetimos estos pasos para instalar java en las VM restantes.

Envío de Archivos

Para enviar archivos desde nuestro ordenador a la VM nos conectamos via sftp con el comando sftp usuario@ip pública y la contraseña de la VM como se muestra en la imagen 17.

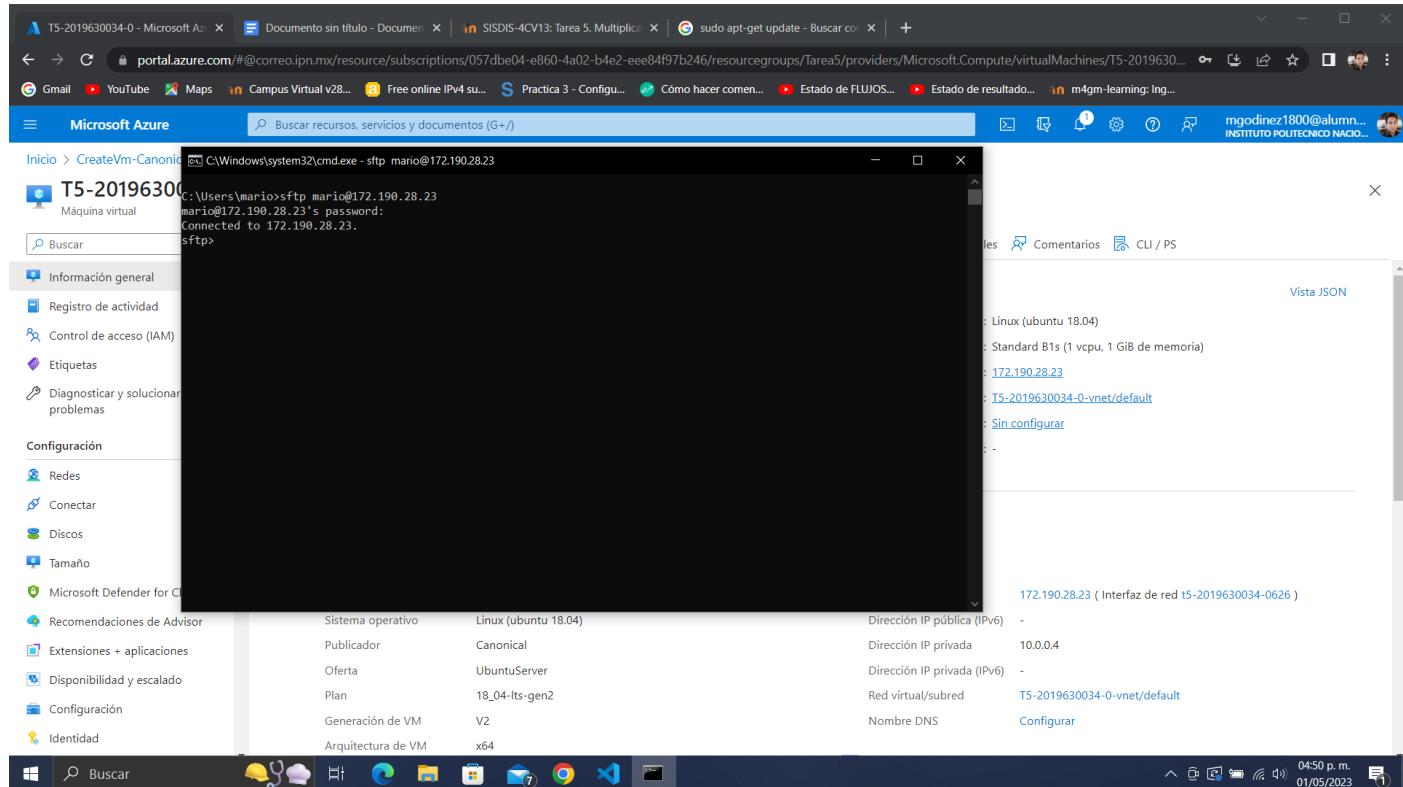


Imagen 17

Una vez nos conectamos via sftp, navegamos hasta los archivos que queremos enviar con el comando lcd para movernos entre carpetas localmente como se muestra en la imagen 18.

The screenshot shows a Microsoft Azure virtual machine interface. On the left, the Azure portal sidebar is visible with options like 'Información general', 'Configuración', and 'Discos'. The main area is a terminal window titled 'T5-201963034-0 - Microsoft Az' showing an SFTP session. The command history includes:

```
sftp> lcd OneDrive
sftp> ll
sftp> lcd Tarea5
```

The terminal also displays the contents of the 'OneDrive' directory:

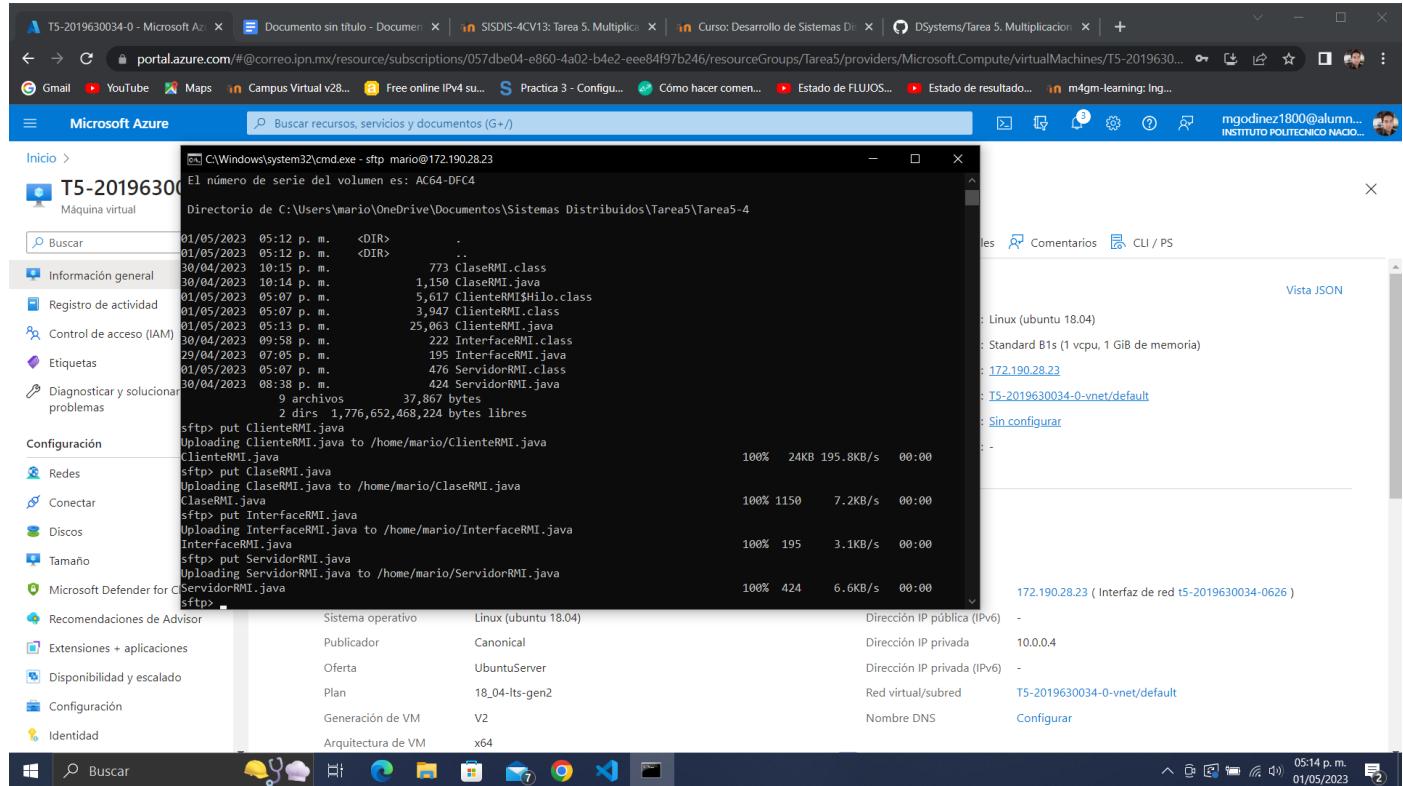
Nombre	Tamaño
Almacén personal.lnk	1,140 bytes
Datos adjuntos	0 bytes
Documento.docx	0 bytes
Documentos	0 bytes
Escritorio	0 bytes
Imágenes	0 bytes
Introducción a OneDrive.pdf	1,041,013 bytes
Música	0 bytes
Presentación 1.pptx	0 bytes
Presentación.pptx	0 bytes
Público	0 bytes
5 archivos	1,042,153 bytes
8 dirs	1,776,650,969,088 bytes libres

At the bottom right of the terminal window, there is a context menu with options: 'Seleccionar todo', 'No seleccionar ninguno', and 'Invertir selección'. The status bar at the bottom right shows the date and time: '05/05 p. m. 01/05/2023'.

Imagen 18

Una vez estamos en el directorio, los archivos que se van a enviar para el nodo 0 (Cliente RMI) son ClienteRMI, ClaseRMI, InterfazRMI y el Servidor RMI. Para el nodo 1 y 2 solamente enviaremos los archivos Servidor RMI, InterfazRMI y ClaseRMI.

Para mandar los archivos utilizamos el comando put <nombreArchivo> como se muestra en la imagen 19.



```
C:\Windows\system32\cmd.exe - sftp mario@172.190.28.23
El numero de serie del volumen es: AC64-DFC4
Directorio de C:\Users\mario\OneDrive\Documentos\Sistemas Distribuidos\Tarea5\Tarea5-4
01/05/2023 05:12 p. m. <DIR> .
01/05/2023 05:12 p. m. <DIR> ..
30/04/2023 10:15 p. m. 773 ClaseRMI.class
30/04/2023 10:14 p. m. 1,150 ClaseRMI.java
01/05/2023 05:07 p. m. 5,617 ClienteRMISHiilo.class
01/05/2023 05:07 p. m. 3,947 ClienteRMI.class
01/05/2023 05:13 p. m. 25,063 ClienteRMI.java
30/04/2023 09:58 p. m. 222 InterfaceRMI.class
29/04/2023 07:05 p. m. 195 InterfaceRMI.java
01/05/2023 05:07 p. m. 476 ServidorRMI.class
30/04/2023 08:38 p. m. 424 ServidorRMI.java
9 archivos 37,867 bytes
2 dirs 1,776,652,468,224 bytes libres
sftp> put ClienteRMI.java
Uploading ClienteRMI.java to /home/mario/ClienteRMI.java
ClienteRMI.java                                         100% 24KB 195.8KB/s 00:00
sftp> put ClaseRMI.java
Uploading ClaseRMI.java to /home/mario/ClaseRMI.java
ClaseRMI.java                                         100% 1150    7.2KB/s 00:00
sftp> put InterfaceRMI.java
Uploading InterfaceRMI.java to /home/mario/InterfaceRMI.java
InterfaceRMI.java                                     100% 195    3.1KB/s 00:00
sftp> put ServidorRMI.java
Uploading ServidorRMI.java to /home/mario/ServidorRMI.java
ServidorRMI.java                                     100% 424    6.6KB/s 00:00
sftp>
```

Imagen 19

Para verificar que los archivos se encuentren en la VM ejecutamos el comando ls como se muestra en la imagen 20.

The screenshot shows a Microsoft Azure interface with a terminal window open. The terminal window title is "Seleccionar C:\Windows\system32\cmd.exe - sftp mario@172.190.28.23". The command "ls" is run, showing a directory listing of Java class files. Below the terminal, there's a file transfer progress bar for "sftp> put ClienteRMI.java" and other files. The Azure sidebar shows the VM configuration, including network settings and disk details. The status bar at the bottom right indicates the IP address 172.190.28.23 and the timestamp 05:15 p.m. 01/05/2023.

```
01/05/2023 05:12 p. m. <DIR> .
01/05/2023 05:12 p. m. <DIR> ..
30/04/2023 10:15 p. m. 773 ClaseRMI.class
01/05/2023 05:07 p. m. 1,150 ClaseRMI.java
01/05/2023 05:07 p. m. 5,617 ClienteRMIServicio.class
01/05/2023 05:07 p. m. 3,947 ClienteRMI.class
01/05/2023 05:13 p. m. 25,063 ClienteRMI.java
30/04/2023 09:58 p. m. 222 InterfaceRMI.class
29/04/2023 07:05 p. m. 195 InterfaceRMI.java
01/05/2023 05:07 p. m. 476 ServidorRMI.class
30/04/2023 08:38 p. m. 424 ServidorRMI.java
9 archivos 37,867 bytes
2 dirs 1,776,652,468,224 bytes libres
```

```
sftp> put ClienteRMI.java
Uploading ClienteRMI.java to /home/mario/ClienteRMI.java
ClienteRMI.java
sftp> put ClaseRMI.java
Uploading ClaseRMI.java to /home/mario/ClaseRMI.java
ClaseRMI.java
sftp> put InterfaceRMI.java
Uploading InterfaceRMI.java to /home/mario/InterfaceRMI.java
InterfaceRMI.java
sftp> put ServidorRMI.java
Uploading ServidorRMI.java to /home/mario/ServidorRMI.java
ServidorRMI.java
sftp> ls
```

ClaseRMI.java	ClienteRMI.java	InterfaceRMI.java	ServidorRMI.java
sistema operativo	Linux (ubuntu 18.04)	Dirección IP pública (IPv6)	-
publicador	Canonical	Dirección IP privada	10.0.0.4
oferta	UbuntuServer	Dirección IP privada (IPv6)	-
plan	18_04-lts-gen2	Red virtual/subred	T5-201963034-0-vnet/default
generación de VM	V2	Nombre DNS	Configurar
arquitectura de VM	x64		

Imagen 20

Y ya se tienen los archivos en la VM. Para los demás nodos 1 y 2 ejecutamos los mismos pasos pero solamente mandamos los archivos ServidorRMI, ClaseRMI y InterfaceRMI.

Nodo1

Ponemos los archivos ServidorRMI, ClaseRMI y InterfaceRMI como se muestra en la imagen 21.

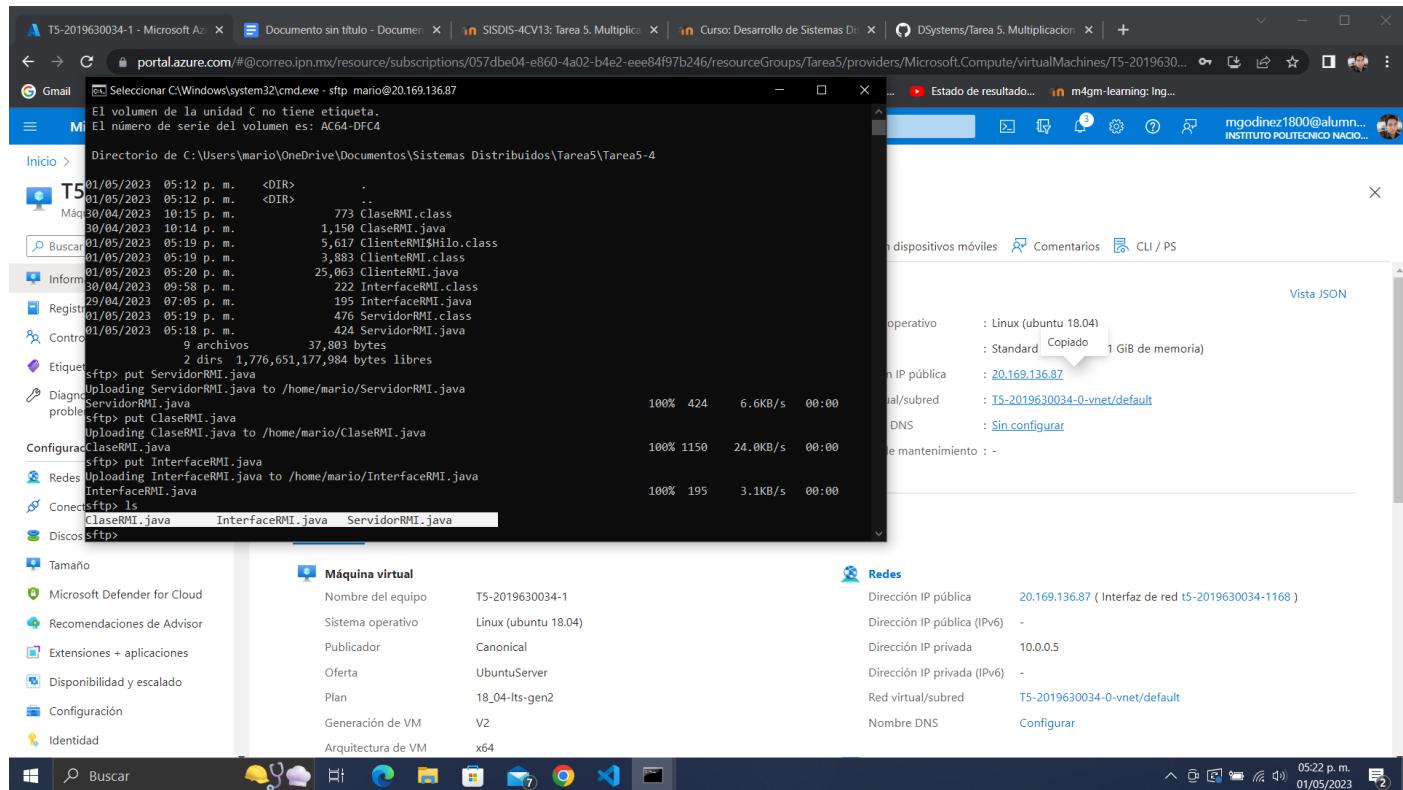


Imagen 21

Nodo2

Ponemos los archivos ServidorRMI, ClaseRMI y InterfaceRMI como se muestra en la imagen 23.

The screenshot shows a Microsoft Windows desktop environment with several open windows:

- Microsoft Edge:** Multiple tabs are open, including portal.azure.com, Documento sin título - Documento, SISDIS-4CV13: Tarea 5. Multiplicac..., Curso: Desarrollo de Sistemas Di..., DSysystems/Tarea 5. Multiplicación, and m4gm-learning: Ing... .
- File Transfer:** A terminal window titled "T5-2019630034-2 - Microsoft Az..." shows the command "sftp mario@20.124.183.91". It lists three files being uploaded:
 - ServidorRMI.java (100% 424 5.3KB/s 00:00)
 - ClaseRMI.java (100% 1150 14.4KB/s 00:00)
 - InterfaceRMI.java (100% 195 3.0KB/s 00:00)
- Terminal Session:** Another terminal window shows the command "sftp> ll". The output lists the contents of the directory:

```
01/05/2023 05:12 p. m. <DIR> .
01/05/2023 05:12 p. m. <DIR> ..
773 ClaseRMI.class
1,158 ClaseRMI.java
5,617 ClienteRMI$Hilo.class
3,883 ClienteRMI.class
25,063 ClienteRMI.java
222 InterfaceRMI.class
195 InterfaceRMI.java
476 ServidorRMI.class
424 ServidorRMI.java
9 archivos 37,803 bytes
2 dirs 1,776,660,471,888 bytes libres
```
- File Explorer:** Shows the local directory C:\Users\mario\OneDrive\Documentos\Sistemas Distribuidos\Tarea5\Tarea5-4.
- System Information:** Shows details about the VM, including its name (T5-2019630034-2), IP address (20.124.183.91), and configuration details like plan (18_04_Its-gen2) and architecture (x64).
- Taskbar:** Standard Windows taskbar with icons for Start, Search, Task View, and various pinned applications.

Imagen 23.

De esta manera ya tenemos los archivos necesarios en los nodos para la ejecución del programa

Ejecución del programa

Para correr el programa primero compilamos los 3 archivos de Nodo 1 y 2 como se muestra en las imágenes 24 y 25.

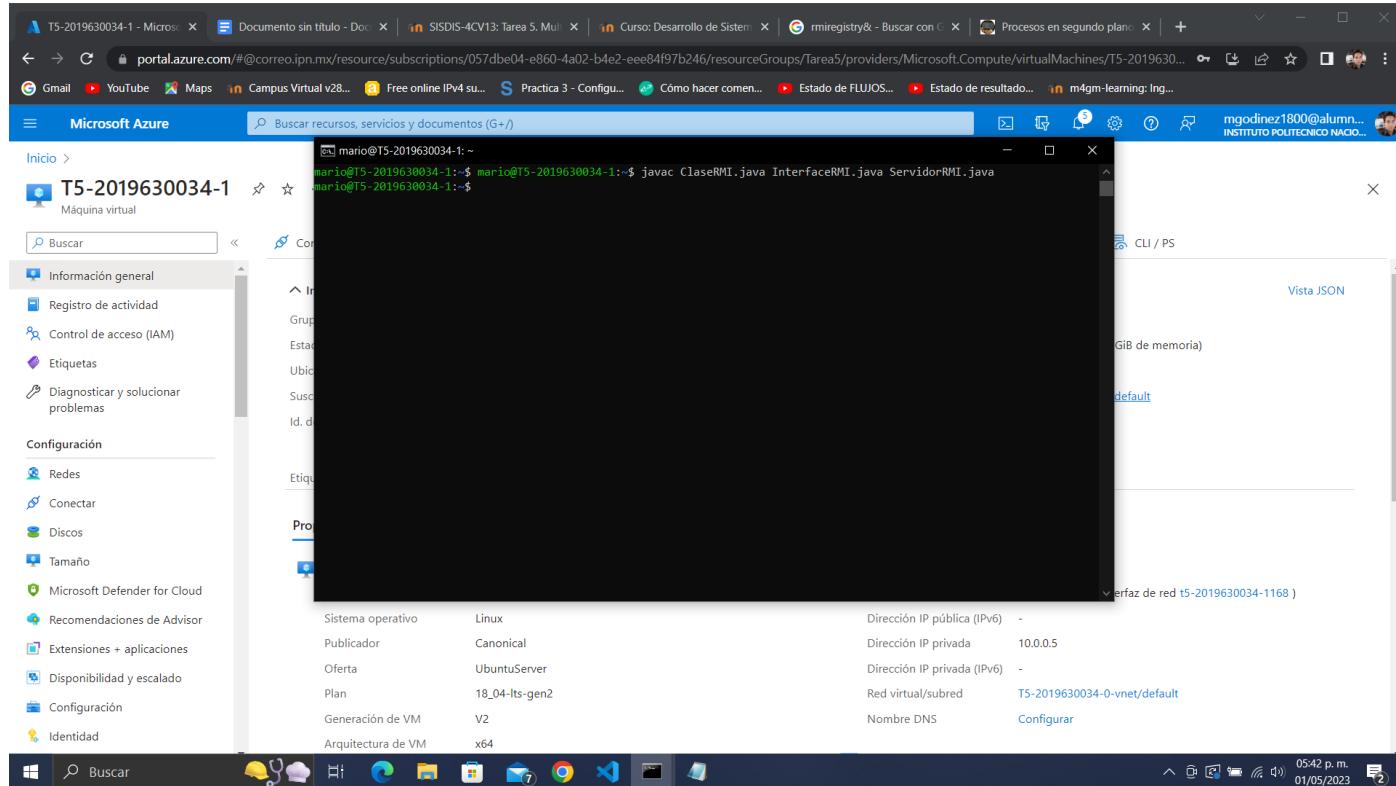


Imagen 24. Compilación de los programas en Nodo1

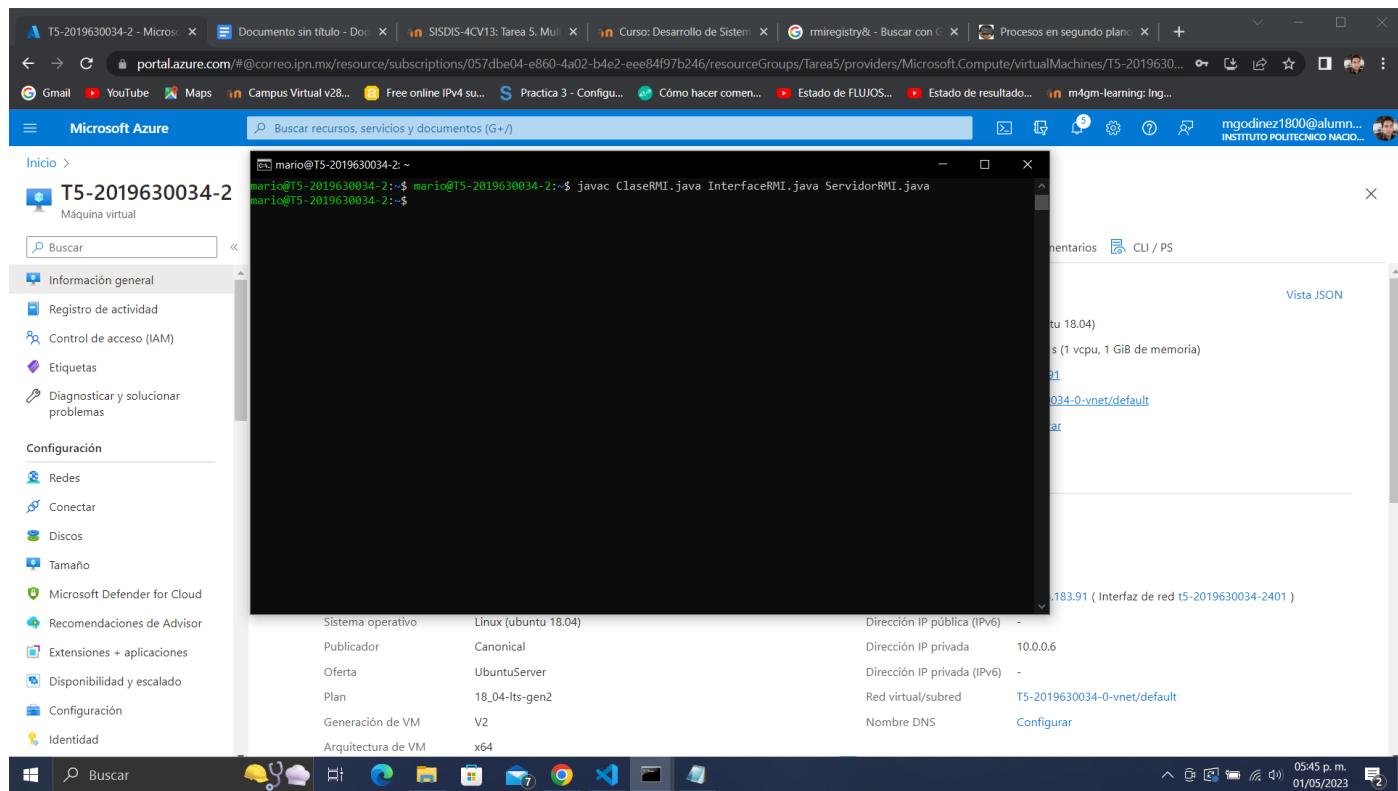


Imagen 25. Compilación de programas en Nodo2

Luego se ejecuta el rmiregistry con el comando rmiregistry& en ambos nodos, se le agrega un & al final de la instrucción para ejecutar el comando en segundo plano debido a que seguiremos ejecutando instrucciones en las terminales como se muestra en las imágenes 26 y 27.

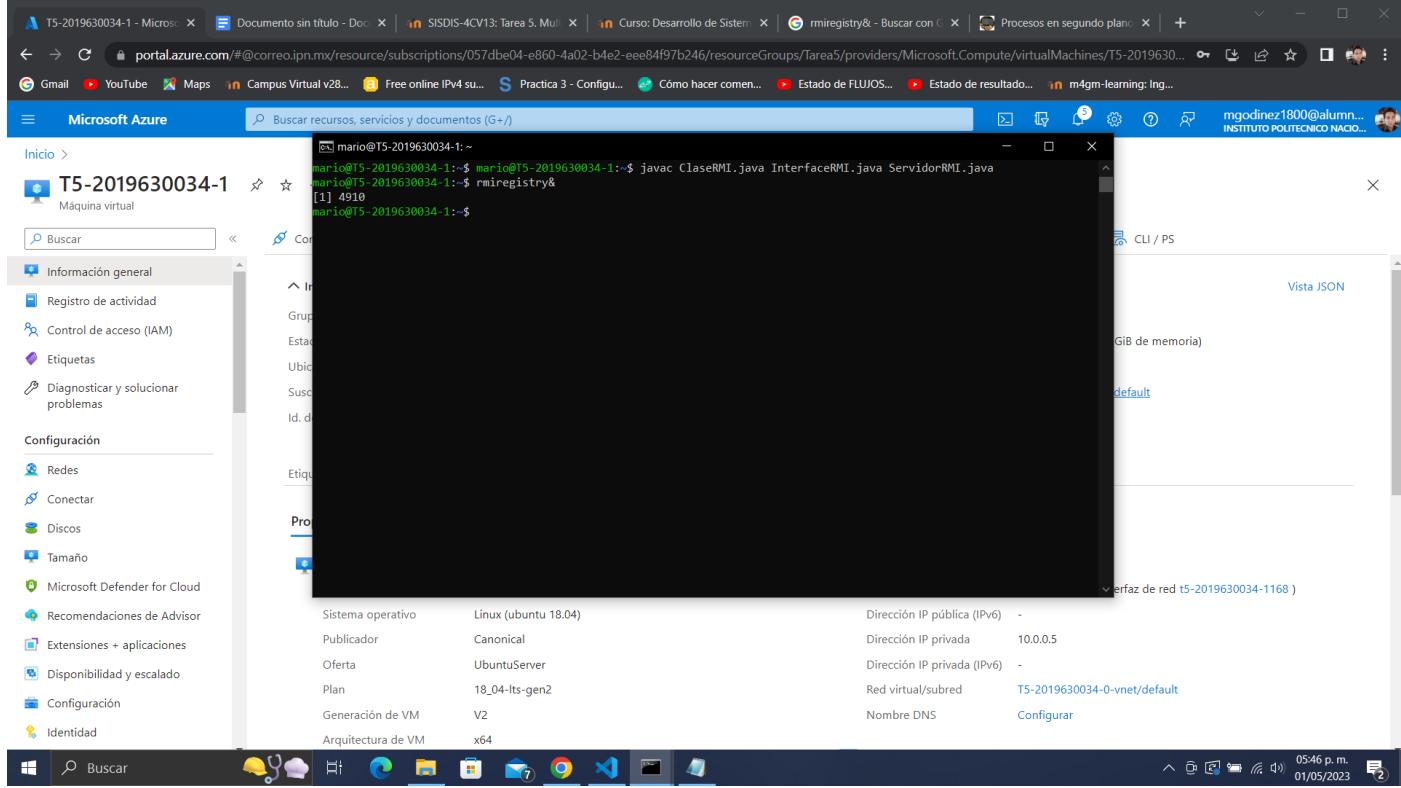


Imagen 27. Ejecución del rmiregistry en el Nodo1

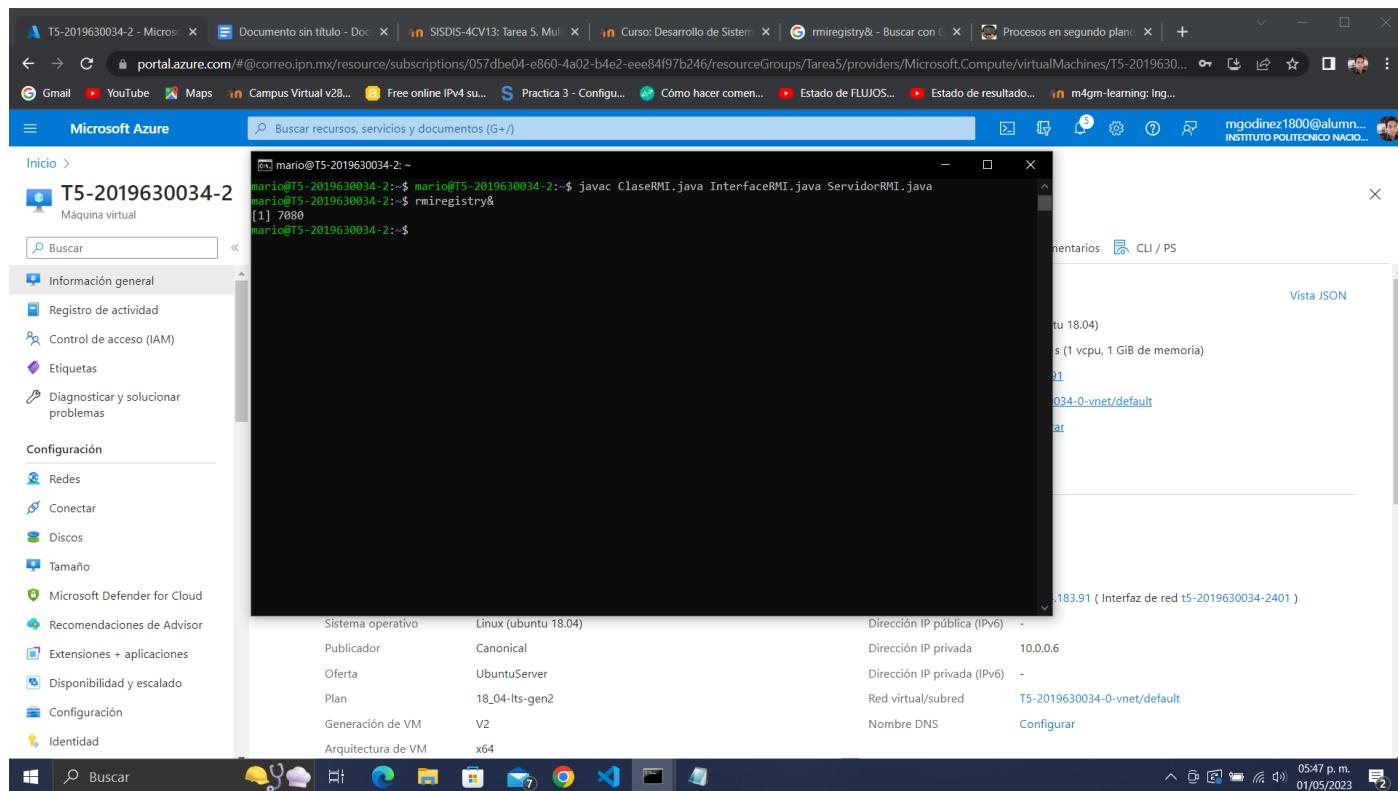


Imagen 28. Ejecución del rmiregistry en el nodo2

Finalmente se ejecuta el ServidorRMI en Nodo 1 y 2 como se muestra en las imágenes 29 y 30.

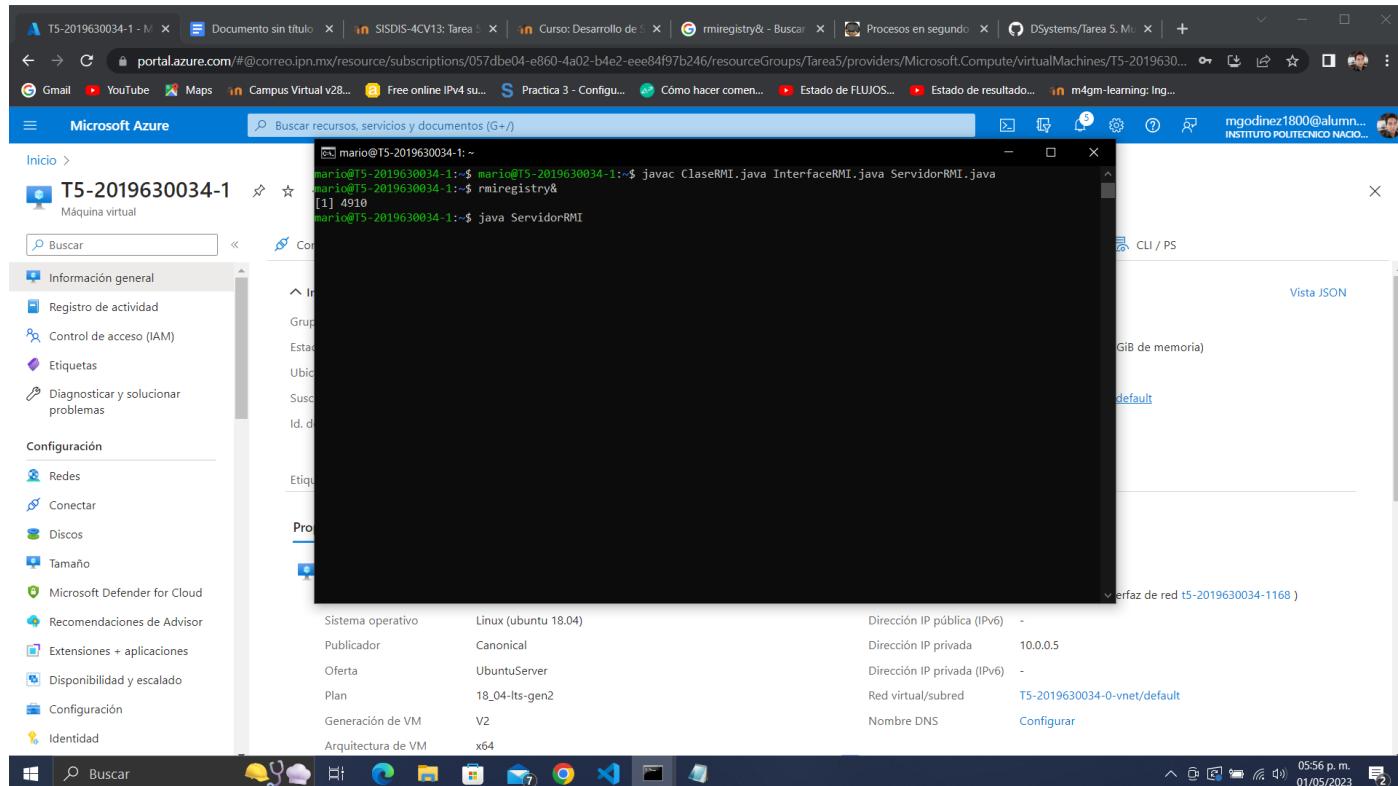


Imagen 29. Ejecución del Servidor RMI en Nodo 1

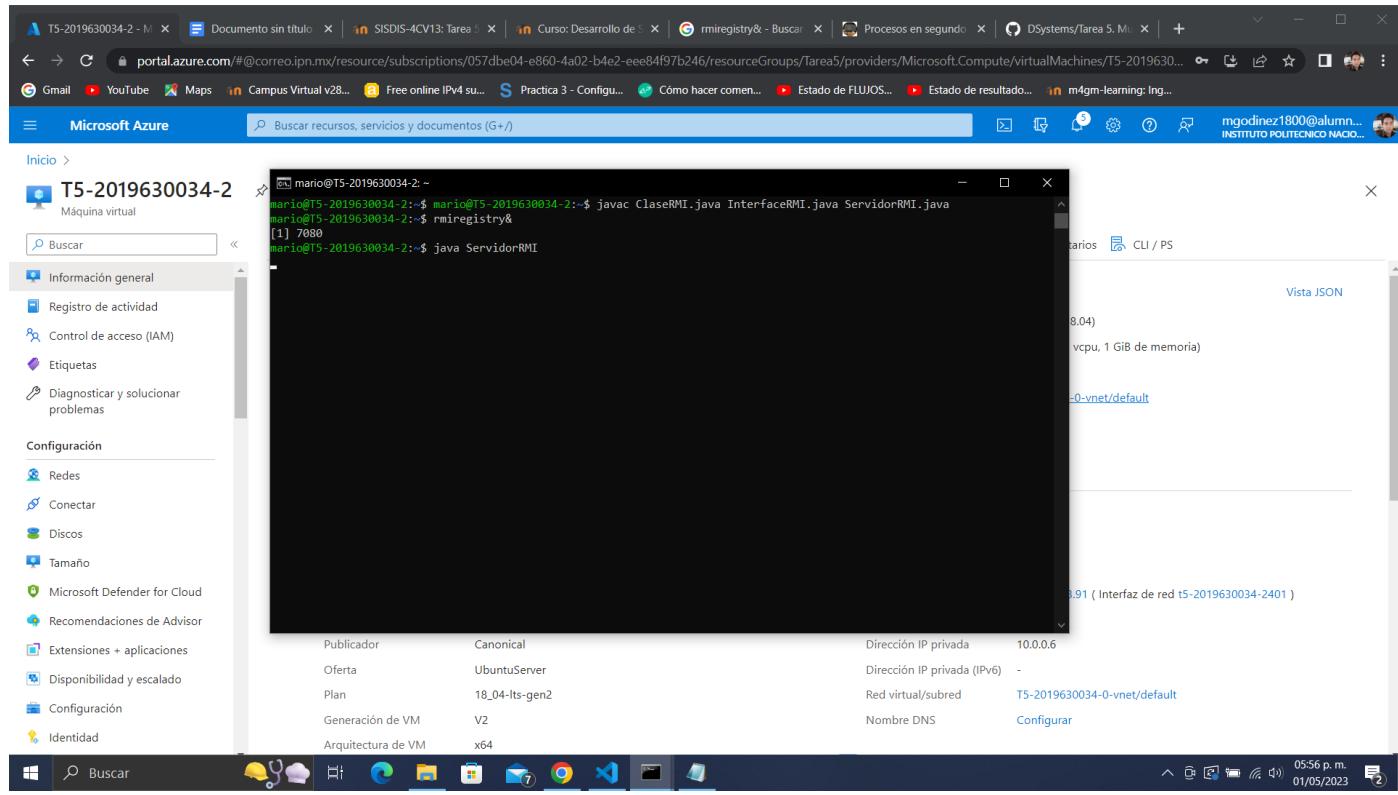


Imagen 30. Ejecución del ServidorRMI en Nodo 2

Luego compilamos los archivos de nodo 0 y de igual manera ejecutamos el rmiregistry y el ServidorRMI en el nodo0 como se muestra en la imagen 31.

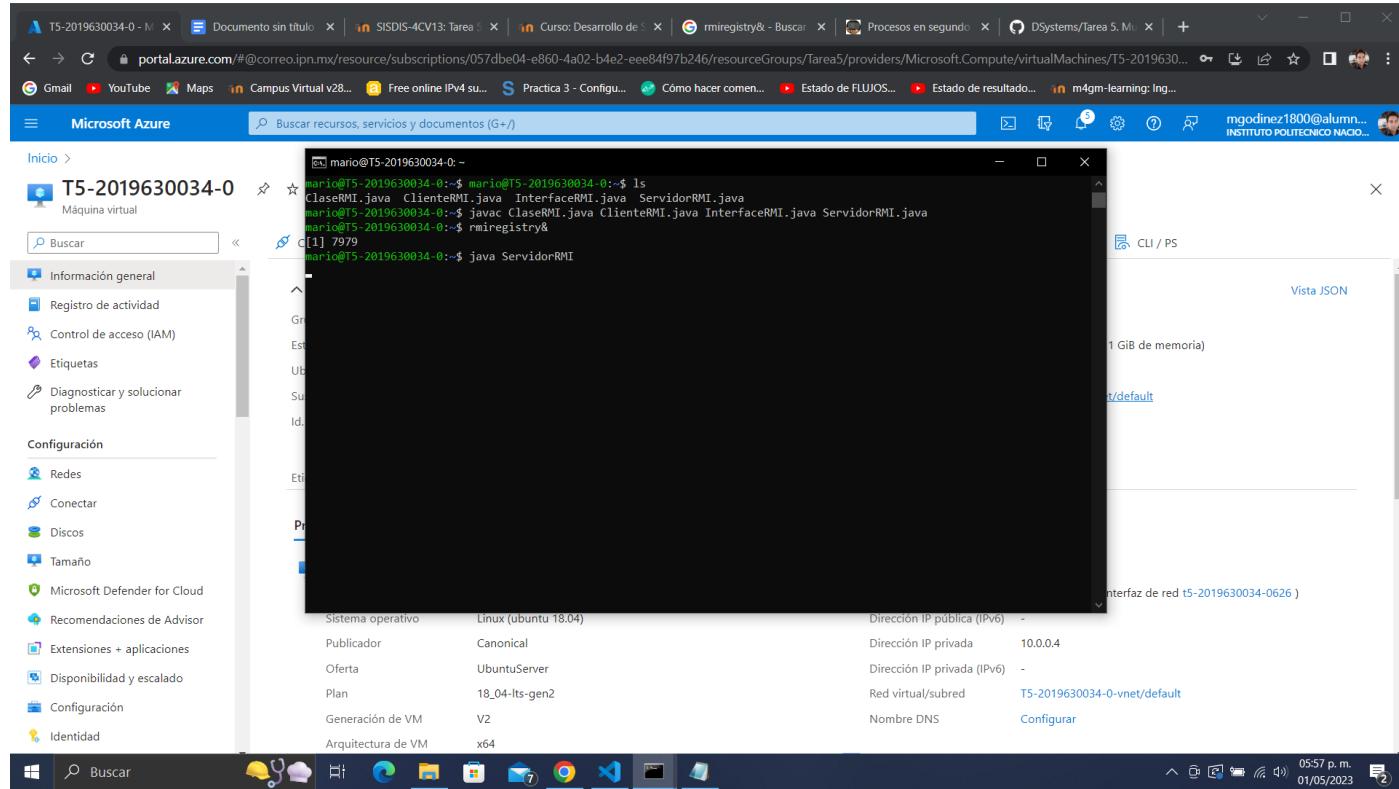


Imagen 31. Compilación, ejecución de rmiregistry y ServidorRMI en Nodo0

Ejecución para N = 9 y M = 4

Una vez tenemos los servidores y el rmiregistry corriendo en cada uno de los nodo, abrimos otra terminal en Nodo0 (conectándose de nuevo por ssh) y corremos el programa con el comando java <nombre del programa> porque ya se ha compilado antes y el resultado es:

```
mario@T5-2019630034-0:~$ mario@T5-2019630034-0:~$ java ClienteRMI
Matriz C:
126.0 90.0 54.0 18.0 -18.0 -54.0 -90.0 -126.0 -162.0
162.0 110.0 58.0 6.0 -46.0 -98.0 -150.0 -202.0 -254.0
198.0 130.0 62.0 -6.0 -74.0 -142.0 -210.0 -278.0 -346.0
234.0 150.0 66.0 -18.0 -102.0 -186.0 -270.0 -354.0 -438.0
270.0 170.0 70.0 -30.0 -138.0 -228.0 -320.0 -420.0 -520.0
306.0 190.0 74.0 -42.0 -158.0 -274.0 -390.0 -506.0 -622.0
342.0 210.0 78.0 -54.0 -186.0 -318.0 -450.0 -582.0 -714.0
378.0 230.0 82.0 -66.0 -214.0 -362.0 -510.0 -658.0 -806.0
414.0 250.0 86.0 -78.0 -242.0 -406.0 -570.0 -734.0 -898.0
Ubicación:
El CheckSum es:-10530.0
mario@T5-2019630034-0:~$
```

Imagen 32

Como se observa en la imagen 32, nos muestra la matriz resultante C y el checksum = -10530.0

Para N = 900 y M = 400

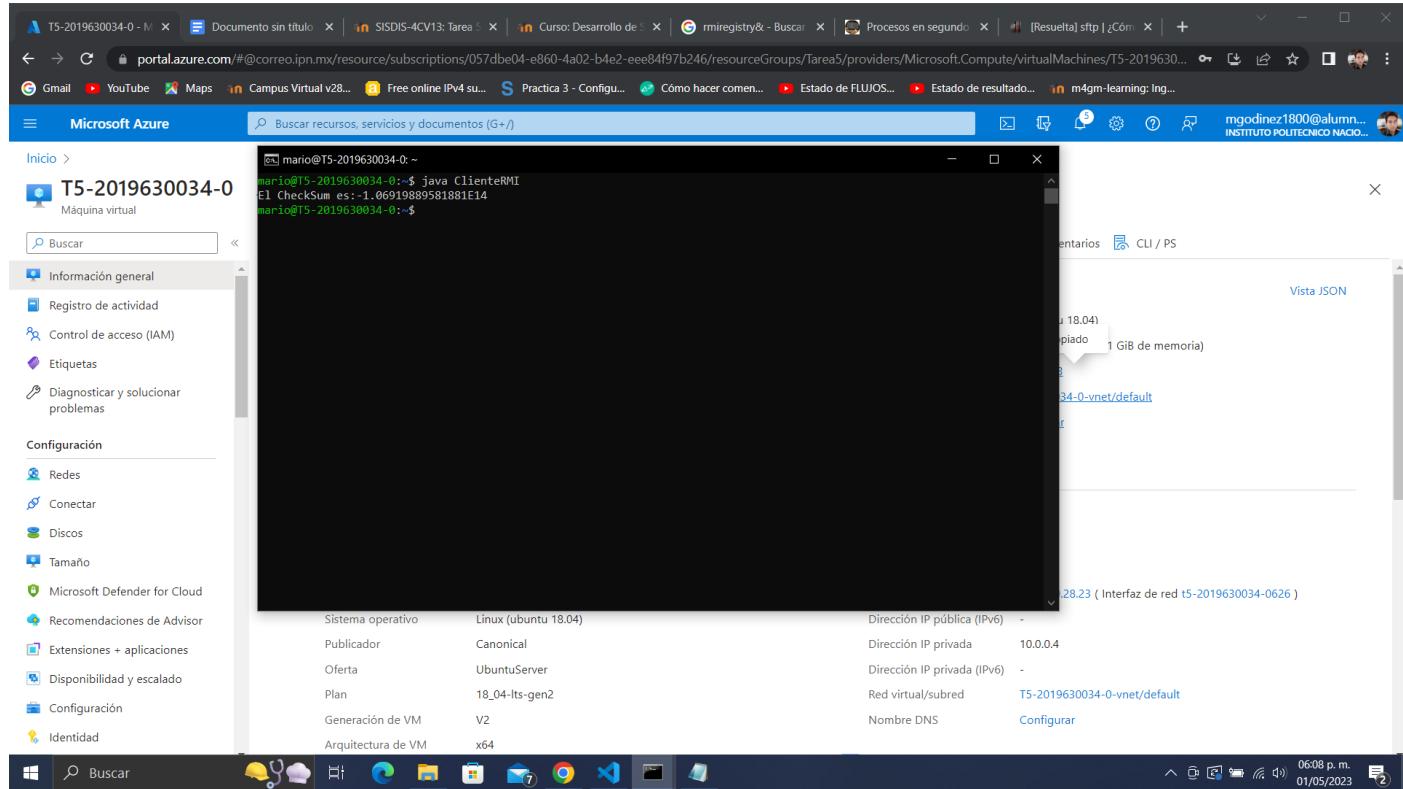


Imagen 33

Como se observa en la imagen 33, solamente nos muestra el checksum = -1.06919889581881E14

Conversación sostenida con ChatGPT

Para abordar esta práctica primero se le pidió al chat de manera modular que primero escribiera la inicialización de las matrices A y B, posteriormente se le pidió al chat obtener la matriz transpuesta de B Bt.

Mario Sebastian

hola!

ChatGPT

¡Hola! ¿En qué puedo ayudarte hoy?

Mario Sebastian

necesito un programa en java que inicialice 2 matrices A de dimensiones NxM y B de dimensiones MxN, los elementos de A y B deberán ser de tipo float y se inicializan de la siguiente manera: $A[i][j] = 2 * i + 3 * j$ y $B[i][j] = 3 * i - 2 * j$ donde A[i][j] y B[i][j] son elementos de A y B respectivamente

ChatGPT

¡Hola! Aquí te dejo un ejemplo de programa en Java que inicializa dos matrices A y B según las especificaciones que has dado:

```
public class Matrices {  
    public static void main(String[] args) {  
        int n = 4; // número de filas de A y número de columnas de B  
        int m = 3; // número de columnas de A y número de filas de B  
  
        // inicialización de la matriz A  
        float[][] A = new float[n][m];  
        for (int i = 0; i < n; i++) {  
            for (int j = 0; j < m; j++) {  
                A[i][j] = 2 * i + 3 * j;  
            }  
        }  
  
        // inicialización de la matriz B  
        float[][] B = new float[m][n];  
        for (int i = 0; i < m; i++) {  
            for (int j = 0; j < n; j++) {  
                B[i][j] = 3 * i - 2 * j;  
            }  
        }  
  
        // impresión de las matrices A y B  
        System.out.println("Matriz A:");  
        for (int i = 0; i < n; i++) {  
            for (int j = 0; j < m; j++) {  
                System.out.print(A[i][j] + " ");  
            }  
        }  
    }  
}
```

```

        }
        System.out.println();
    }

    System.out.println("Matriz B:");
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            System.out.print(B[i][j] + " ");
        }
        System.out.println();
    }
}

```

Este programa crea dos matrices A y B de dimensiones NxM y MxN respectivamente y las inicializa según las fórmulas que has proporcionado. Luego, imprime las matrices en la consola.

Espero que te sea útil. ¡Que tengas un buen día!

Mario Sebastian

gracias, ahora vamos a modificar el programa para obtener Bt que es la transpuesta de la matriz B, solo dame el fragmento de código para convertir la transpuesta de B

ChatGPT

¡Claro! Aquí te dejo el fragmento de código necesario para obtener la transpuesta de la matriz B:

```

// cálculo de la matriz transpuesta de B
float[][] Bt = new float[n][m];
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        Bt[i][j] = B[j][i];
    }
}

// impresión de la matriz transpuesta de B
System.out.println("Matriz B transpuesta:");
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        System.out.print(Bt[i][j] + " ");
    }
    System.out.println();
}

```

Este código crea una matriz Bt de dimensiones MxN y la llena con los elementos de la transpuesta de B, es decir, intercambia las filas por columnas. Luego, imprime la matriz transpuesta de B en la consola.

Espero que te sea útil. ¡Que tengas un buen día!

Mario Sebastian

ahora vamos a dividir las matrices A y Bt en nueve partes, cada parte dividida de la matriz A se almacenará en una matriz A1, A2, ..., A9 y lo mismo para Bt: Bt1, Bt2, ..., Bt9. Solo dame la parte en que hace dicha parte por favor

ChatGPT

¡Por supuesto! Aquí está el fragmento de código para dividir las matrices A y Bt en nueve partes:

```
// definición del tamaño de cada submatriz
int subSizeN = n/3; // número de filas de cada submatriz
int subSizeM = m/3; // número de columnas de cada submatriz

// división de la matriz A en submatrices A1, A2, ..., A9
float[][][] ASub = new float[3][3][subSizeN][subSizeM];
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        for (int k = 0; k < subSizeN; k++) {
            for (int l = 0; l < subSizeM; l++) {
                ASub[i][j][k][l] = A[i*subSizeN + k][j*subSizeM + l];
            }
        }
    }
}

// división de la matriz Bt en submatrices Bt1, Bt2, ..., Bt9
float[][][] BtSub = new float[3][3][subSizeM][subSizeN];
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        for (int k = 0; k < subSizeM; k++) {
            for (int l = 0; l < subSizeN; l++) {
                BtSub[i][j][k][l] = Bt[j*subSizeM + k][i*subSizeN + l];
            }
        }
    }
}
```

Este código primero define el tamaño de cada submatriz, que es la tercera parte del tamaño de la matriz original en cada dimensión. Luego, utiliza cuatro bucles anidados para dividir la matriz A en submatrices A1, A2, ..., A9 y la matriz Bt en submatrices Bt1, Bt2, ..., Bt9.

El índice i y el índice j recorren las filas y las columnas de las submatrices ASub y BtSub, mientras que los índices k y l recorren las filas y las columnas de las submatrices en la matriz original A y la matriz transpuesta de B.

Espero que esto te ayude. Si tienes alguna pregunta, no dudes en preguntar. ¡Que tengas un buen día!

Nota: Al final se unieron las partes del código y se hicieron pequeñas modificaciones

Posteriormente ya que se tiene la parte de inicializar matrices, obtener la transpuesta de B y la división en 9 partes de de A y Bt ahora pasé a la parte de la multiplicación de 2 matrices en una nueva conversación.

Mario Sebastian

hola!

ChatGPT

¡Hola! ¿En qué puedo ayudarte hoy?

Mario Sebastian

necesito un método en java que multiplique 2 matrices A y Bt

ChatGPT

¡Por supuesto! Aquí tienes un método en Java que multiplica dos matrices A y B^t:

```
public static double[][] multiplyMatrices(double[][] A, double[][] B) {  
    if (A[0].length != B.length) {  
        throw new IllegalArgumentException("La dimensión de las matrices no es compatible  
para la multiplicación");  
    }  
    double[][] Bt = transposeMatrix(B);
```

```
    int m = A.length;  
    int n = Bt.length;  
    double[][] result = new double[m][n];  
    for (int i = 0; i < m; i++) {  
        for (int j = 0; j < n; j++) {  
            double sum = 0.0;  
            for (int k = 0; k < A[0].length; k++) {  
                sum += A[i][k] * Bt[j][k];  
            }  
            result[i][j] = sum;  
        }  
    }  
    return result;
```

```
}
```

```
public static double[][] transposeMatrix(double[][] matrix) {
```

```
    int m = matrix.length;  
    int n = matrix[0].length;  
    double[][] result = new double[n][m];  
    for (int i = 0; i < m; i++) {  
        for (int j = 0; j < n; j++) {  
            result[j][i] = matrix[i][j];  
        }  
    }  
    return result;
```

```
}
```

Este método toma dos matrices A y B como entrada, y devuelve su producto $A * B^t$. Primero se verifica que las dimensiones de las matrices sean compatibles para la multiplicación, luego se transpone la matriz B para poder multiplicarla por A. Luego, se realiza la multiplicación matricial y se almacena el resultado en la matriz result. Finalmente, se devuelve el resultado.

Nota: Algo interesante es que infiere que B^t es una matriz transpuesta así que la calculó en el código pero en nuestro caso ya teníamos la matriz transpuesta y no hacia falta calcularla nuevamente.

Mario Sebastian

la matriz B^t ya está transpuesta, no hace falta hacerlo

ChatGPT

¡Tienes razón! Si la matriz B ya está transpuesta, entonces no es necesario transponerla nuevamente. Puedes usar este método en su lugar:

```
public static double[][] multiplyMatrices(double[][] A, double[][] Bt) {  
    if (A[0].length != Bt[0].length) {  
        throw new IllegalArgumentException("La dimensión de las matrices no es compatible  
        para la multiplicación");  
    }  
    int m = A.length;  
    int n = Bt.length;  
    double[][] result = new double[m][n];  
    for (int i = 0; i < m; i++) {  
        for (int j = 0; j < n; j++) {  
            double sum = 0.0;  
            for (int k = 0; k < A[0].length; k++) {  
                sum += A[i][k] * Bt[j][k];  
            }  
            result[i][j] = sum;  
        }  
    }  
    return result;  
}
```

Este método es muy similar al anterior, pero utiliza la matriz B^t en lugar de su transpuesta. Recuerda que la dimensión de la matriz B^t debe ser $(n \times m)$ en lugar de $(m \times n)$ para poder ser multiplicada por A.

Mario Sebastian

perfecto, solo que las matrices son de tipo float, lo puedes modificar por favor?

ChatGPT

¡Claro que sí! Aquí está el código actualizado para trabajar con matrices de tipo float:

```
public static float[][] multiplyMatrices(float[][] A, float[][] Bt) {  
    if (A[0].length != Bt[0].length) {  
        throw new IllegalArgumentException("La dimensión de las matrices no es compatible  
        para la multiplicación");  
    }
```

```

    }
    int m = A.length;
    int n = Bt.length;
    float[][] result = new float[m][n];
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            float sum = 0.0f;
            for (int k = 0; k < A[0].length; k++) {
                sum += A[i][k] * Bt[j][k];
            }
            result[i][j] = sum;
        }
    }
    return result;
}

```

Este código es muy similar al código anterior, pero utiliza tipos de datos float en lugar de double. Las variables 0.0 y 0.0f se utilizan para inicializar las variables sum y result respectivamente.

Nota: En este punto se tuvieron intercambiar filas por columnas en la multiplicación debido a que se tiene que hacer renglon por renglon

Una vez teníamos la multiplicación de matrices, procedemos a implementar java RMI

Mario Sebastian

perfecto, ahora si quiero hacer ese mismo método añadiendo java RMI y dicho método de multiplicar matrices se ejecutarán en 3 nodos o máquinas diferentes, como queda el código?

ChatGPT

Para añadir Java RMI al código y ejecutar el método de multiplicar matrices en 3 nodos diferentes, necesitarás realizar varios pasos:

- 1.Crea una interfaz RMI que contenga la definición del método de multiplicación de matrices. Por ejemplo:

```

import java.rmi.Remote;
import java.rmi.RemoteException;

```

```

public interface MatrixMultiplication extends Remote {
    float[][] multiplyMatrices(float[][] A, float[][] Bt) throws RemoteException;
}

```

- 2.Implementa la interfaz RMI en una clase que contenga el método de multiplicación de matrices. Por ejemplo:

```

import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

```

```

public class MatrixMultiplicationImpl extends UnicastRemoteObject implements
MatrixMultiplication {

```

```

public MatrixMultiplicationImpl() throws RemoteException {
    super();
}

public float[][] multiplyMatrices(float[][] A, float[][] B) throws RemoteException {
    // Aquí va el código que definimos anteriormente
}
}

```

3. En cada nodo o máquina que deseas utilizar, crea una instancia de la clase MatrixMultiplicationImpl y regístrala en el registro RMI. Por ejemplo:

```

import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

```

```

public class Node1 {
    public static void main(String[] args) throws Exception {
        MatrixMultiplicationImpl multiplicationImpl = new MatrixMultiplicationImpl();
        Registry registry = LocateRegistry.createRegistry(1099);
        registry.bind("matrixMultiplication", multiplicationImpl);
    }
}

```

Nota que estamos registrando la instancia de MatrixMultiplicationImpl en el puerto 1099 y con el nombre "matrixMultiplication". Debes repetir este paso en cada uno de los nodos que deseas utilizar.

4. En un cuarto programa Java, crea una instancia del objeto remoto de cada nodo utilizando LocateRegistry.getRegistry() y llama al método multiplyMatrices en cada uno de ellos. Por ejemplo:

```

import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

```

```

public class Client {
    public static void main(String[] args) throws Exception {
        Registry registry1 = LocateRegistry.getRegistry("node1_ip", 1099);
        Registry registry2 = LocateRegistry.getRegistry("node2_ip", 1099);
        Registry registry3 = LocateRegistry.getRegistry("node3_ip", 1099);

        MatrixMultiplication multiplication1 = (MatrixMultiplication)
registry1.lookup("matrixMultiplication");
        MatrixMultiplication multiplication2 = (MatrixMultiplication)
registry2.lookup("matrixMultiplication");
        MatrixMultiplication multiplication3 = (MatrixMultiplication)
registry3.lookup("matrixMultiplication");

        // Ahora puedes llamar al método multiplyMatrices en cada objeto remoto
        float[][] result1 = multiplication1.multiplyMatrices(A1, B1);
        float[][] result2 = multiplication2.multiplyMatrices(A2, Bt2);
        float[][] result3 = multiplication3.multiplyMatrices(A3, Bt3);
    }
}

```

```
    }  
}
```

Nota que en este ejemplo estamos llamando al método `lookup()` en cada registro RMI para obtener una instancia del objeto remoto `MatrixMultiplicationImpl` que registramos previamente en cada nodo. También estamos llamando al método `multiplyMatrices` en cada uno de los objetos remotos para realizar la multiplicación de matrices.

Espero que esto te ayude a entender cómo utilizar Java RMI para ejecutar el método de multiplicación de matrices en múltiples nodos. Ten en cuenta que esta es solo una guía básica y que debes adaptarla a

Nota: Al final se acabaron los tokens y ya no pudo continuar con la explicación del código, sin embargo lo más importante si lo escribió y explicó. Al final se adecuó el ejemplo que el chat proporcionó a la práctica.

Una vez tenemos la multiplicación utilizando objetos remotos, solamente nos faltaba la parte de los hilos para procesar en paralelo en los diferentes nodos, para esto continue la conversación anterior.

Mario Sebastian

ahora si quiero llamar al metodo multiplyMatrices en paralelo usando hilos, que debo hacer?

ChatGPT

Para llamar al método `multiplyMatrices` en paralelo usando hilos, puedes crear varios hilos, cada uno de los cuales se encarga de llamar al método en una parte diferente de las matrices de entrada y luego combinar los resultados.

Por ejemplo, supongamos que tienes dos matrices de tamaño $n \times m$ y $m \times p$, y quieres ejecutar el método `multiplyMatrices` en dos hilos diferentes. Podrías crear dos hilos, uno que multiplique las filas 0 a $n/2 - 1$ de la matriz A con todas las columnas de la matriz Bt, y otro que multiplique las filas $n/2$ a $n-1$ de la matriz A con todas las columnas de la matriz Bt. Luego, puedes combinar los resultados de ambos hilos para obtener la matriz de resultados completa.

Aquí hay un ejemplo de cómo podrías hacerlo:

```
public class MatrixMultiplicationThread extends Thread {  
    private float[][] A;  
    private float[][] Bt;  
    private float[][] result;  
    private int startRow;  
    private int endRow;  
  
    public MatrixMultiplicationThread(float[][] A, float[][] Bt, float[][] result, int startRow, int  
endRow) {  
        this.A = A;  
        this.Bt = Bt;  
        this.result = result;  
        this.startRow = startRow;
```

```

        this.endRow = endRow;
    }

    public void run() {
        for (int i = startRow; i < endRow; i++) {
            for (int j = 0; j < Bt.length; j++) {
                float sum = 0;
                for (int k = 0; k < A[i].length; k++) {
                    sum += A[i][k] * Bt[j][k];
                }
                result[i][j] = sum;
            }
        }
    }
}

public static float[][] multiplyMatrices(float[][] A, float[][] Bt) {
    int n = A.length;
    int p = Bt.length;
    float[][] result = new float[n][p];
    int mid = n / 2;

    MatrixMultiplicationThread thread1 = new MatrixMultiplicationThread(A, Bt, result, 0, mid);
    MatrixMultiplicationThread thread2 = new MatrixMultiplicationThread(A, Bt, result, mid, n);

    thread1.start();
    thread2.start();

    try {
        thread1.join();
        thread2.join();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

return result;
}

```

En este ejemplo, hemos creado una clase `MatrixMultiplicationThread` que extiende la clase `Thread`. Esta clase se encarga de realizar la multiplicación de matrices en una parte de las matrices de entrada y almacenar el resultado en una parte correspondiente de la matriz de salida. La clase `MatrixMultiplicationThread` toma como entrada las matrices de entrada, la matriz de salida, y los índices de fila de inicio y fin que especifican la parte de las matrices de entrada que se deben multiplicar.

En el método `multiplyMatrices`, primero inicializamos la matriz de resultados y calculamos el índice de fila en el medio de la matriz de entrada. Luego, creamos dos instancias de `MatrixMultiplicationThread`, una para calcular las filas 0 a `mid-1` y otra para calcular las filas `mid` a `n-1`. Luego, iniciamos los dos hilos y esperamos a que ambos final

Mario Sebastian

recuerda que estamos usando RMI

ChatGPT

Cierto, para utilizar RMI, en lugar de ejecutar la multiplicación de matrices en hilos locales, deberás ejecutarla en hilos remotos utilizando la interfaz remota de RMI.

Primero, deberás definir la interfaz remota que define el método multiplyMatrices que quieras llamar de forma remota. La interfaz debe extender la interfaz Remote y el método debe declarar la excepción RemoteException en su firma, de la siguiente manera:

```
import java.rmi.Remote;
import java.rmi.RemoteException;
```

```
public interface MatrixMultiplicationInterface extends Remote {
    float[][] multiplyMatrices(float[][] A, float[][] Bt) throws RemoteException;
}
```

Luego, en el servidor que ejecuta la multiplicación de matrices, deberás crear una implementación de la interfaz MatrixMultiplicationInterface. La implementación debe extender la clase UnicastRemoteObject para que se pueda registrar como objeto remoto en el registro RMI, de la siguiente manera:

```
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
```

```
public class MatrixMultiplicationImpl extends UnicastRemoteObject implements
MatrixMultiplicationInterface {
    public MatrixMultiplicationImpl() throws RemoteException {
        super();
    }
}
```

```
    public float[][] multiplyMatrices(float[][] A, float[][] Bt) throws RemoteException {
        // Implementación del método de multiplicación de matrices
    }
}
```

En este ejemplo, la implementación de multiplyMatrices es la misma que la que escribimos anteriormente.

Luego, en el cliente que llama al método de multiplicación de matrices, deberás obtener una referencia al objeto remoto registrado en el registro RMI y llamar al método remoto multiplyMatrices en un hilo remoto, utilizando el objeto java.rmi.Naming para obtener la referencia al objeto remoto.

Aquí hay un ejemplo de cómo podrías hacerlo:

```
import java.rmi.Naming;
import java.rmi.RemoteException;
```

```
public class Client {
    public static void main(String[] args) throws RemoteException {
        try {
```

```

MatrixMultiplicationInterface matrixMultiplication = (MatrixMultiplicationInterface)
Naming.lookup("//localhost/MatrixMultiplication");

float[][] A = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
float[][] Bt = {{1, 2, 3}, {4, 5, 6}};
int n = A.length;
int p = Bt.length;

float[][] result = new float[n][p];

MatrixMultiplicationThread thread1 = new
MatrixMultiplicationThread(matrixMultiplication, A, Bt, result, 0, n / 2);
MatrixMultiplicationThread thread2 = new
MatrixMultiplicationThread(matrixMultiplication, A, Bt, result, n / 2, n);

thread1.start();
thread2.start();

try {
    thread1.join();
    thread2.join();
} catch (InterruptedException e) {
    e.printStackTrace();
}

// Imprimir el resultado de la multiplicación de matrices
for (int i = 0; i < n; i++) {
    for (int j = 0; j < p; j++) {
        System.out.print(result[i][j] + " ");
    }
    System.out.println();
}
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

En este ejemplo, hemos creado dos hilos MatrixMultiplicationThread, cada uno de los cuales llama al método multiplyMatrices en un objeto remoto de la

Nota: Al final se quedó sin tokens pero lo más importante y el código estaba completo.

Al final se adecuaron los ejemplos de los hilos al código

Conclusiones

A diferencia de la práctica 2, en esta práctica se implementó RMI en vez de paso de mensajes, note que los cálculos fueron más rápidos debido a que se hicieron en paralelo, a diferencia de la práctica 2 que en el peor de los casos se tardó 4 min en hacer los cálculos, en esta práctica solo tomo aprox 2 segundos. Algo de mayor valor es que lo probamos en la nube y así le da más realismo a la práctica debido a que hacemos transferencias de archivos a equipos remotamente y ejecutamos todo remotamente en otros equipos en vez de usar localhost. La importancia de Java RMI es que ya no debemos pasar directamente los métodos en cada nodo si no solamente las matrices y los valores que se requieran para llevar a cabo los cálculos en los nodos del sistema además de que es más rápido todo.