



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Desarrollo de Sistemas Distribuidos

“Tarea 2: Multiplicación de matrices distribuida
utilizando el paso de mensajes”

Grupo: 4CV13

Alumno:

Godinez Morales Mario Sebastian

Profesor:

Pineda Guerrero Carlos

Desarrollo

Creación de la primera máquina virtual (Nodo1) T2-2019630034-1

La imagen 1 muestra la página principal de azure, aquí seleccionamos la parte de máquinas virtuales:

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with several tabs and links. Below it is a search bar labeled "Buscar recursos, servicios y documentos (G+J)". The main content area is titled "Servicios de Azure" and features a grid of service icons: "Crear un recurso" (plus sign), "Máquinas virtuales" (server icon), "Centro de inicio rápido" (rocket icon), "App Services" (globe icon), "Cuentas de almacenamiento" (bar chart icon), "SQL Database" (SQL icon), "Azure Cosmos DB" (globe with lightning icon), "Servicios de Kubernetes" (cube icon), "Aplicación de funciones" (lightning bolt icon), and "Más servicios" (arrow icon). Below this is a section titled "Recursos" with two tabs: "Reciente" (selected) and "Favorito". It lists two resources: "T2-2019630034-1" (Máquina virtual) and "Tarea2" (Grupo de recursos). A "Ver todo" link is also present. Further down are sections for "Navegar" (with links to "Suscripciones", "Grupos de recursos", "Todos los recursos", and "Panel") and "Herramientas" (with links to "Microsoft Learn", "Azure Monitor", "Microsoft Defender for Cloud", and "Administración de costos"). The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray.

Imagen 1

Posteriormente se selecciona la información básica del recurso a crear, esto es asignar un grupo de recursos, definir la zona en donde se alojará dicho recurso, el hardware, SO y contraseñas como lo muestra la imagen 2, 3 y 4:

The screenshot shows the 'Create a virtual machine' wizard on the Microsoft Azure portal. The current step is 'Configure instance details'. The form includes fields for Subscription (selected: 'Azure for Students'), Resource Group ('(Nuevo) Tarea2'), and various configuration options like Name, Region, Availability options, Security type, Image, and Architecture (selected: 'x64'). A warning message at the top states: 'Al cambiar opciones básicas se pueden restablecer las selecciones realizadas. Revise todas las opciones antes de crear la máquina virtual.' (When changing basic options, they can be reset. Please review all options before creating the virtual machine.)

Imagen 2

Curso: Desarrollo de Sistemas Di... x A Crear una máquina virtual - Micro... x SISDIS-4CV13: Tarea 2. Multiplic... x +

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

Gmail YouTube Maps Campus Virtual v28... Free online IPv4 su... Practica 3 - Configu... Cómo hacer comen... Estado de FLUJOS... Estado de resultado... m4gm-learning: Ing...

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

Inicio > Máquinas virtuales >

Crear una máquina virtual ...

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

Tamaño * Standard_B1s - 1 vcpu, 1 GiB de memoria (9.64 US\$/mes) Ver todos los tamaños

Cuenta de administrador

Tipo de autenticación Clave pública SSH Contraseña

Nombre de usuario * mario

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 * SSH (22)

Revisar y crear < Anterior Siguiente: Discos > Enviar comentarios



Imagen 3

Curso: Desarrollo de Sistemas Di... Seleccionar un tamaño de máqui... SISDIS-4CV13: Tarea 2. Multiplic... +

portal.azure.com/#view/Microsoft_Azure_Compute/SpecPickerV2Blade/subscriptionId/057dbe04-e860-4a02-b4e2-eee84f97b246/regionId/switzerlandnorth/extendedRegionId/switzerland...

Gmail YouTube Maps Campus Virtual v28... Free online IPv4 su... Práctica 3 - Configuración Cómo hacer comentar... Estado de FLUJOS... Estado de resultado... m4gm-learning: Ing...

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

Inicio > Máquinas virtuales > Crear una máquina virtual >

Seleccionar un tamaño de máquina virtual ...

Tamaño de VM ↑↓	Tipo ↑↓	vCPU ↑↓	RAM (GiB) ↑↓	Discos de datos ↑↓	E/S máxima por s... ↑↓	Almacenamiento tem... ↑↓	Disco premium ↑↓	Costo/mes ↑↓
Más usados por los usuarios de Azure ↗ Los tamaños más usados por los usuarios en Azure.								
DS1_v2 ↗	Uso general	1	3.5	4	3200	7	Se admite	54,52 US\$
D2s_v3 ↗	Uso general	2	8	4	3200	16	Se admite	96,36 US\$
D2as_v4 ↗	Uso general	2	8	4	3200	16	Se admite	100,01 US\$
B2s ↗	Uso general	2	4	4	1280	8	Se admite	38,54 US\$
B1s ↗	Uso general	1	1	2	320	4	Se admite	9,64 US\$
B2ms ↗	Uso general	2	8	4	1920	16	Se admite	77,09 US\$
B1ls ↗	Uso general	1	0.5	2	320	4	Se admite	4,82 US\$
DS2_v2 ↗	Uso general	2	7	8	6400	14	Se admite	109,21 US\$
B4ms ↗	Uso general	4	16	8	2880	32	Se admite	154,18 US\$
D4s_v3 ↗	Uso general	4	16	8	6400	32	Se admite	192,72 US\$
DS3_v2 ↗	Uso general	4	14	16	12800	28	Se admite	218,42 US\$
D8s_v3 ↗	Uso general	8	32	16	12800	64	Se admite	385,44 US\$
> D-Series v5 Tamaños de la familia D de última generación recomendados para sus necesidades de uso general.								
Seleccionar Prices presented are estimates in USD that include only Azure infrastructure costs and any discounts for the subscription and location. The prices don't include any applicable software costs. Final charges will appear in your local currency in cost analysis and billing views. Vea la calculadora de precios de Azure.								
Enviar comentarios								

Buscar

05:05 p.m. 24/03/2023

Imagen 4

Posteriormente al dar siguiente, seleccionamos el tipo de disco que tendrá nuestra VM como se muestra en la imagen 5

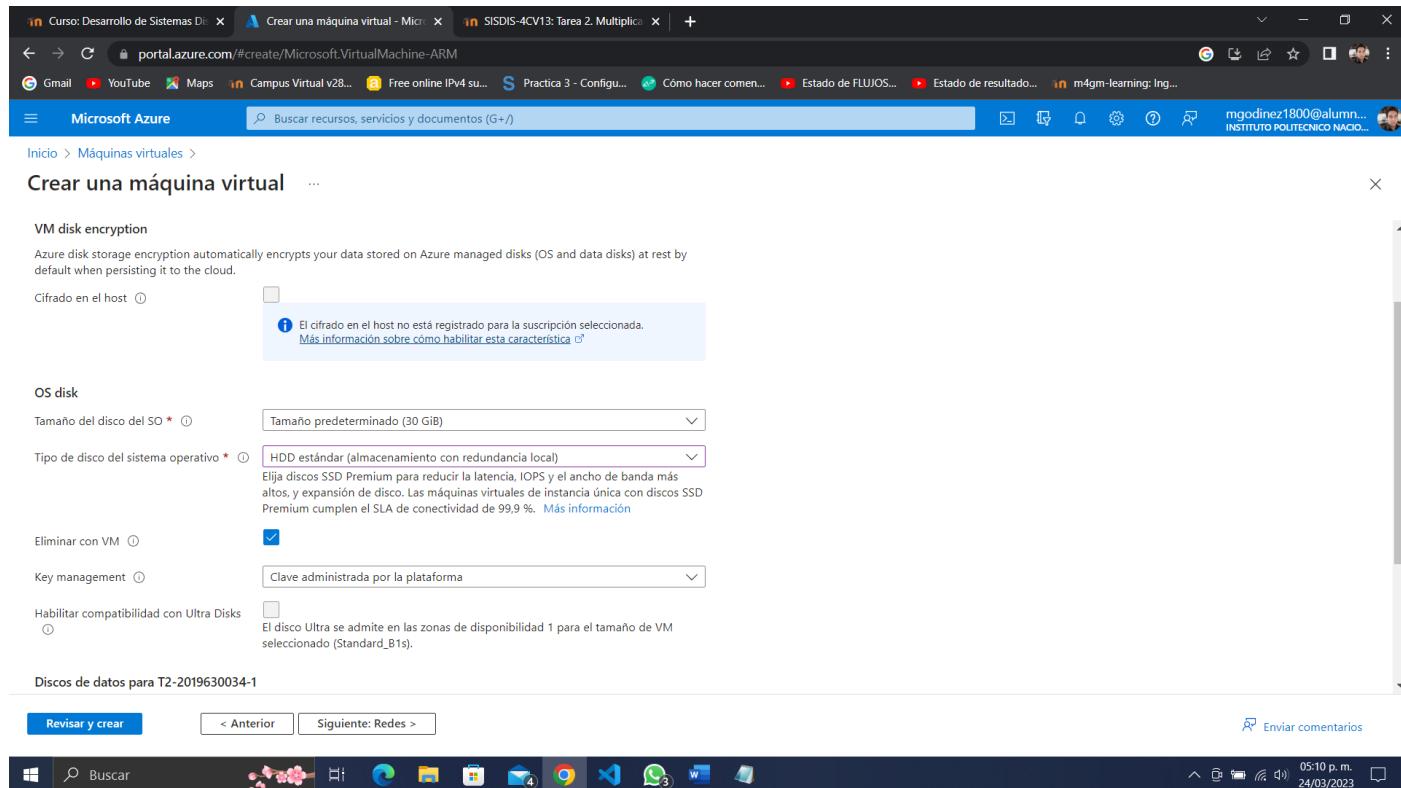


Imagen 5

Después sigue el apartado de la red, en nuestro caso no modifiqué nada

Campo: Desarrollo de Sistemas De... | Crear una máquina virtual - Micr... | SISDIS-4CV13: Tarea 2. Multiplic... | +

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

Gmail YouTube Maps Campus Virtual v28... Free online IPv4 su... Práctica 3 - Configu... Cómo hacer comen... Estado de FLUJOS... Estado de resultado... m4gm-learning: Ing... mgodinez1800@alumn... INSTITUTO POLITÉCNICO NAC...

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

Inicio > Máquinas virtuales > Crear una máquina virtual ...

Datos básicos Discos Redes Administración Monitoring Opciones avanzadas Etiquetas Revisar y crear

Configure la tarjeta de interfaz de red (NIC) a fin de definir la conectividad de red para la máquina virtual. Puede controlar los puertos y la conectividad entrante y saliente con reglas de grupos de seguridad o bien aplicar una solución de equilibrio de carga ya existente. [Más información](#)

Interfaz de red

Al crear una máquina virtual, se crea una interfaz de red automáticamente.

Red virtual * ⓘ (nuevo) T2-2019630034-1-vnet
Crear nuevo

Subred * ⓘ (nuevo) default (10.0.0.0/24)

IP pública ⓘ (nuevo) T2-2019630034-1-ip
Crear nuevo

Grupo de seguridad de red de NIC ⓘ Ninguno Básico Opciones avanzadas

Puertos de entrada públicos * ⓘ Ninguno Permitir los puertos seleccionados

Revisar y crear < Anterior Siguiente: Administración >

Enviar comentarios

Windows Search Buscar

05:10 p. m. 24/03/2023

Imagen 6

Al dar en el botón siguiente, pasamos a la parte de administración como se muestra en la imagen 7, en nuestro caso no se modificó nada

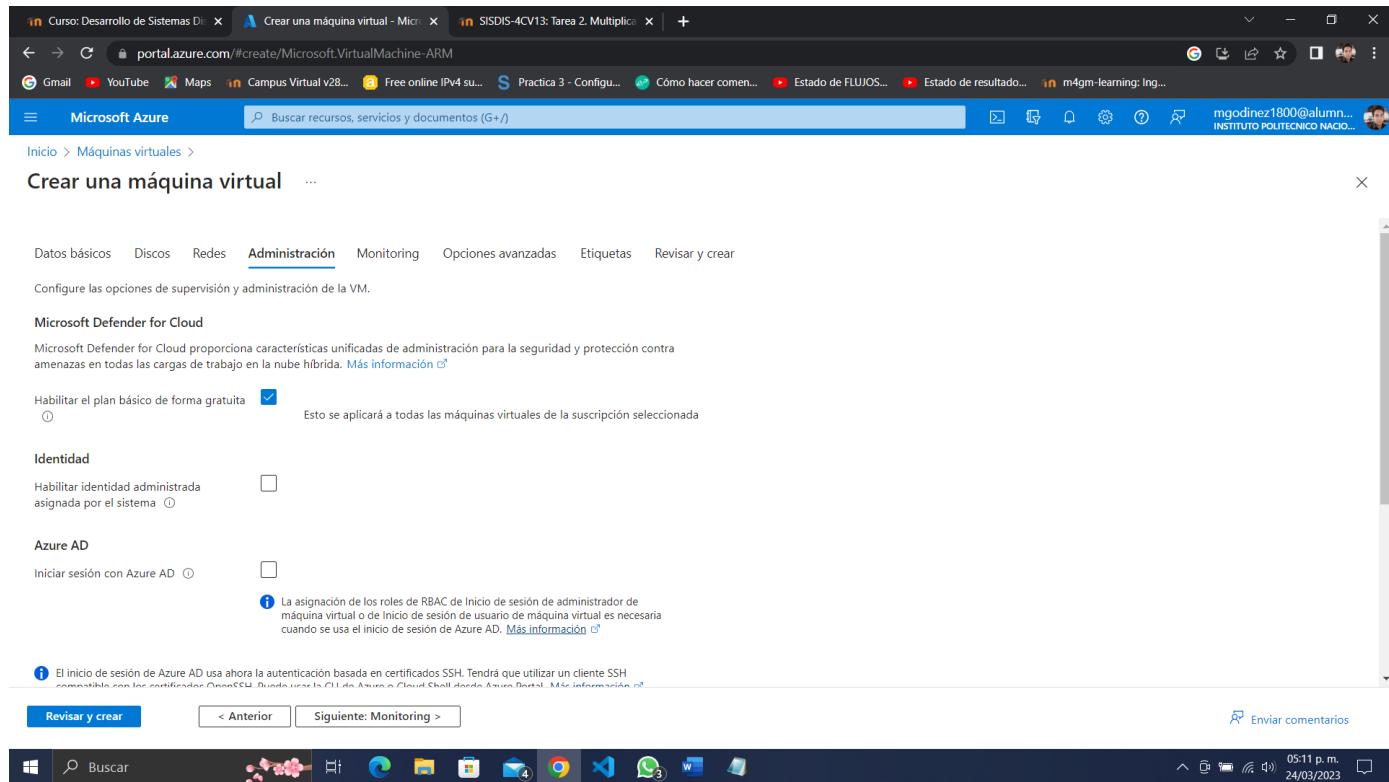


Imagen 7

Al final revisamos las características de nuestra VM y el costo que tendrá por hora, si todo está bien damos en el botón crear como se muestra en la imagen 8

The screenshot shows the Microsoft Azure portal interface for creating a new virtual machine. The browser tab is titled 'Crear una máquina virtual - Microsoft Azure'. The main content area is titled 'Crear una máquina virtual' and shows a green success message: 'Validación superada'. Below this, there are tabs for 'Datos básicos', 'Discos', 'Redes', 'Administración', 'Monitoring', 'Opciones avanzadas', 'Etiquetas', and 'Revisar y crear'. The 'Revisar y crear' tab is currently selected. A note below the tabs states: 'El costo que se indica a continuación es una estimación y no el precio final. Use [Calculadora de precios](#) para todas sus necesidades de precios.' In the 'Price' section, it shows '1 X Standard B1s by Microsoft' with a price of '0,0132 USD/hr'. There is also a link to 'Pricing for other VM sizes'. Below this, there is a 'TERMS' section with a detailed legal statement. At the bottom, there are buttons for 'Crear' (Create), '< Anterior' (Previous), 'Siguiente >' (Next), and 'Descargar una plantilla para la automatización' (Download a template for automation). The status bar at the bottom right shows the date and time: '24/03/2023 05:12 p. m.'

Imagen 8

Una vez tenemos la VM creada, procedemos a hacer la conexión mediante SSH utilizando la terminal de comandos de windows escribiendo ssh usuario@ip pública, una vez ingresados dichos datos, nos pide la contraseña de la VM y si todo es correcto, nos conectaremos vía ssh a la VM como se muestra en la imagen 9 y 10.

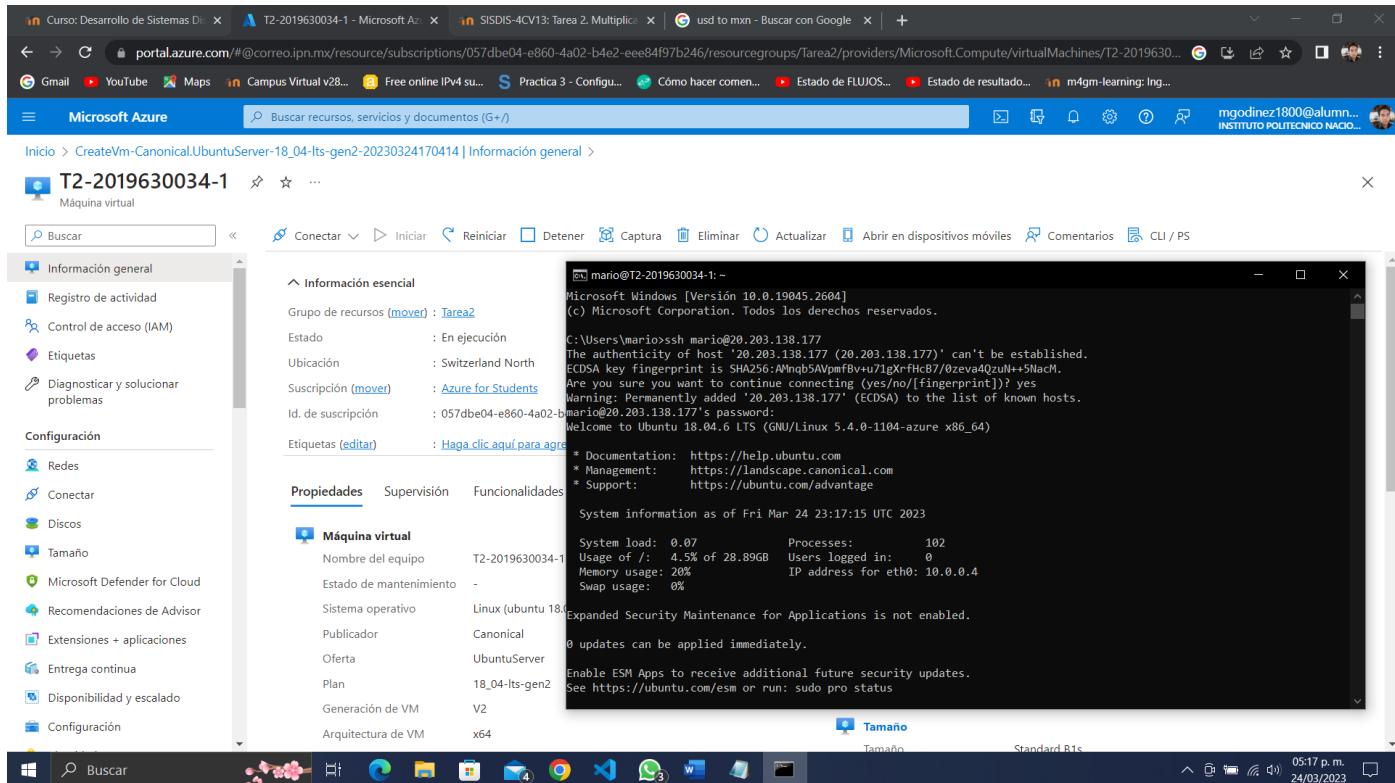


Imagen 9

T2-2019630034-1 | Información general

Información general

Estado: En ejecución

Ubicación: Switzerland North

Suscripción: Azure for Students

Etiquetas: Haga clic aquí para agr...

Propiedades

Máquina virtual

Nombre del equipo	T2-2019630034-1
Estado de mantenimiento	-
Sistema operativo	Linux (ubuntu 18.04)
Publicador	Canonical
Oferta	UbuntuServer
Plan	18_04-lts-gen2
Generación de VM	V2
Arquitectura de VM	x64

Información esencial

System information as of Fri Mar 24 23:17:15 UTC 2023

System load:	0.07	Processes:	102
Usage of /:	4.5% of 28.89GB	Users logged in:	0
Memory usage:	20%	IP address for eth0:	10.0.0.4
Swap usage:	0%	Expanded Security Maintenance for Applications is not enabled.	

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates. See <https://ubuntu.com/esm> or run: sudo pro status

mario@T2-2019630034-1:~

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

To run a command as administrator (user "root"), use "sudo <command>". See "man sudo_root" for details.

mario@T2-2019630034-1:~\$

Imagen 10

Actualizamos listas de paquetes para instalar java en la VM con el comando sudo apt update como se muestra en la imagen 11

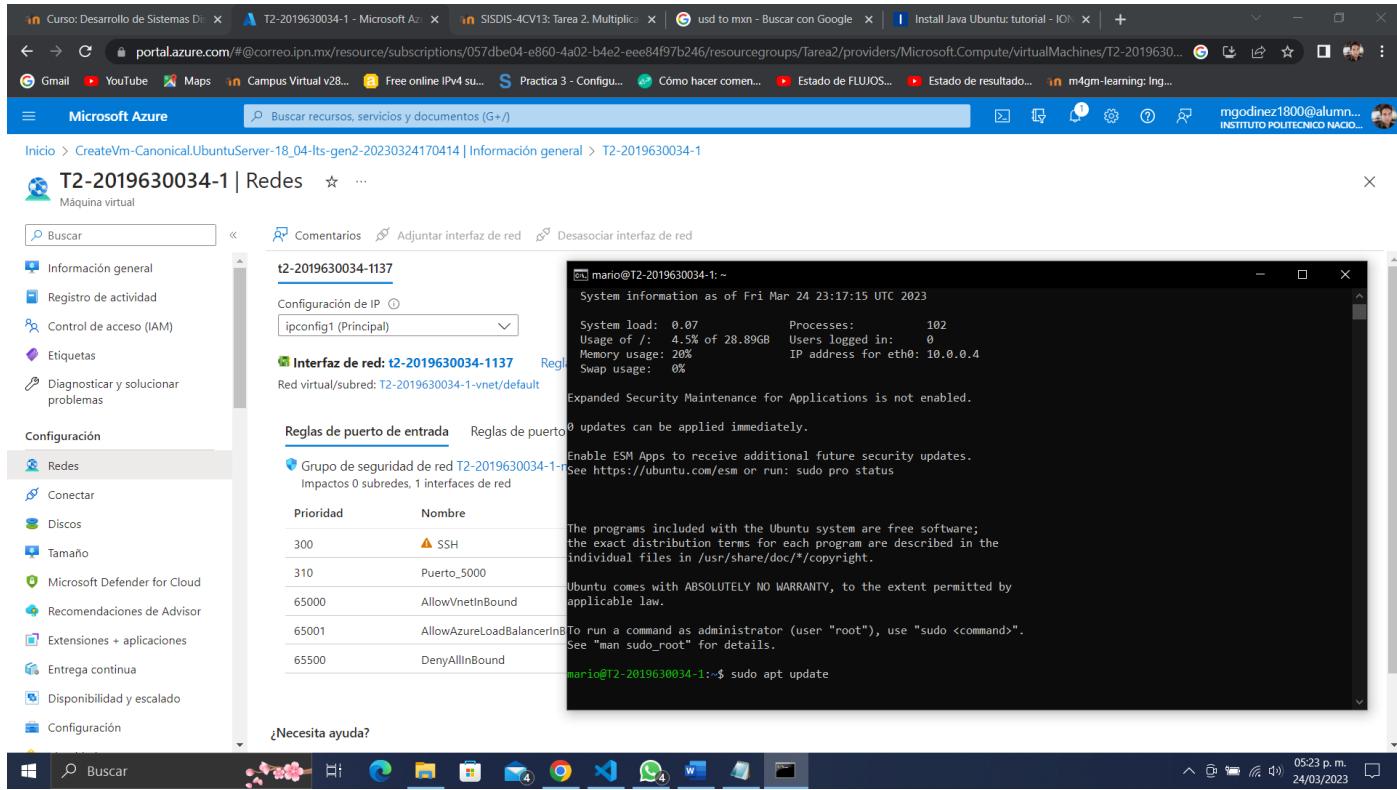


Imagen 11

Instalamos java debido a que no lo tenemos instalado como se muestra en la imagen 12, 13 y 14

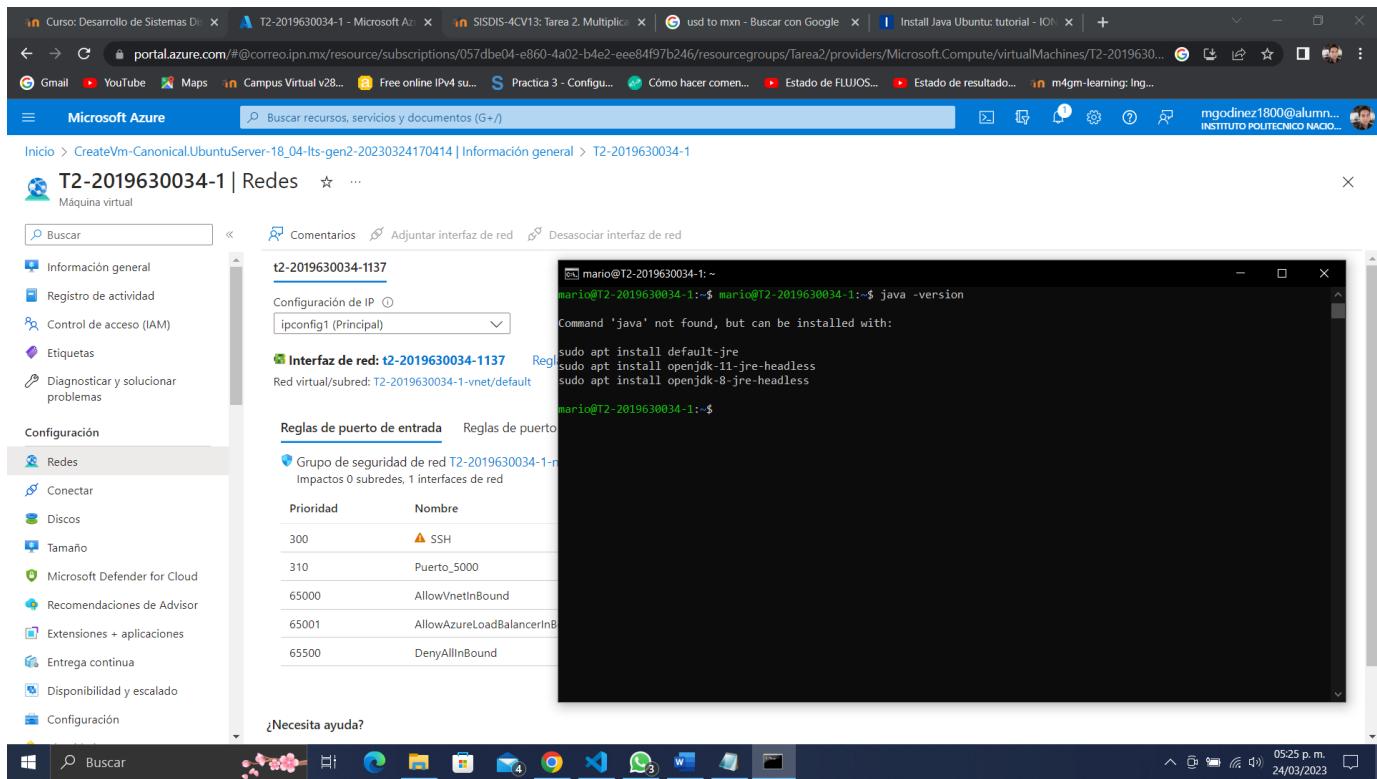


Imagen 12

Instalamos Java

The screenshot shows the Microsoft Azure portal interface. The main title bar says "T2-2019630034-1 | Redes". On the left, there's a sidebar with "Redes" selected under "Configuración". The main pane shows a terminal window with the following text:

```
mario@T2-2019630034-1:~$ mario@T2-2019630034-1:~$ java -version
Command 'java' not found, but can be installed with:
sudo apt install default-jre
sudo apt install openjdk-11-jre-headless
sudo apt install openjdk-8-jre-headless
mario@T2-2019630034-1:~$ sudo apt install default-jre
```

Below the terminal, there's a table titled "Reglas de puerto de entrada" (Inbound port rules) with the following data:

Prioridad	Nombre
300	SSH
310	Puerto_5000
65000	AllowVnetInBound
65001	AllowAzureLoadBalancerInB
65500	DenyAllInBound

Imagen 13

Screenshot of a Microsoft Azure virtual machine interface showing network configuration details.

Header: Microsoft Azure | Buscar recursos, servicios y documentos (G+)

Breadcrumbs: Inicio > CreateVm-CanonicalUbuntuServer-18_04-Its-gen2-20230324170414 | Información general > T2-2019630034-1

Title: T2-2019630034-1 | Redes

Left Sidebar:

- Información general
- Registro de actividad
- Control de acceso (IAM)
- Etiquetas
- Diagnosticar y solucionar problemas
- Configuración

 - Redes
 - Conectar
 - Discos
 - Tamaño
 - Microsoft Defender for Cloud
 - Recomendaciones de Advisor
 - Extensiones + aplicaciones
 - Entrega continua
 - Disponibilidad y escalado
 - Configuración

Central Content:

Redes: t2-2019630034-1137

Configuración de IP: ipconfig1 (Principal)

Interfaz de red: t2-2019630034-1137

Reglas de puerto de entrada:

Prioridad	Nombre
300	SSH
310	Puerto_5000
65000	AllowVNetInBound
65001	AllowAzureLoadBalancerInBound
65500	DenyAllInBound

Log Output:

```
mario@T2-2019630034-1: ~
get:32 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libxcb-dri2-0 amd64 1.13-2-ubuntu18.04 [6920 B]
get:33 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libxcb-dri3-0 amd64 1.13-2-ubuntu18.04 [6568 B]
get:34 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libxcb-glx0 amd64 1.13-2-ubuntu18.04 [22.1 kB]
get:35 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libxcb-present0 amd64 1.13-2-ubuntu18.04 [5592 B]
[...]
get:36 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libxcb-sync1 amd64 1.13-2-ubuntu18.04 [8808 B]
get:37 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 libxdamage1 amd64 1:1.1.4-3 [6934 B]
get:38 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 libxfixes3 amd64 1:5.0.3-1 [16.8 kB]
get:39 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 libxshmfence1 amd64 1.3-1 [5028 B]
get:40 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libdrm-amdgpu1 amd64 2.4.101-2-18.04.1 [18.2 kB]
get:41 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 libdrm-intel1 amd64 2.4.101-2-18.04.1 [60.0 kB]
get:42 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libdrm-nouveau2 amd64 2.4.101-2-18.04.1 [16.5 kB]
[...]
get:43 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libdrm-nouveau2 amd64 2.4.101-2-18.04.1 [17.9 kB]
get:44 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libdrm-radeon1 amd64 2.4.101-2-18.04.1 [21.7 kB]
get:45 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libl1vm10 amd64 1:10.0.0-4ubuntu1~18.04.2 [15.4 kB]
[...]
get:46 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libsensors4 amd64 1:3.4.0-4ubuntu0.1 [28.3 kB]
get:47 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgl1-mesa-dri amd64 20.0.8-0ubuntu1~18.04.1 [933 kB]
get:48 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libglx-mesa0 amd64 20.0.8-0ubuntu1~18.04.1 [139 kB]
get:49 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libglx0 amd64 1.0.0-2ubuntu2.3 [28.1 kB]
get:50 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgl1 amd64 1.0.0-2ubuntu2.3 [86.2 kB]
get:51 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libglf7 amd64 5.1.4-2ubuntu0.1 [30.9 kB]
get:52 http://azure.archive.ubuntu.com/ubuntu bionic/main amd64 libxi6 amd64 2:1.7.9-1 [29.2 kB]
get:53 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 openjdk-11-jre amd64 11.0.18+10ubuntu1~18.04.1 [174 kB]
```

Bottom Bar: Buscar, 05:26 p.m., 24/03/2023

Imagen 14

Finalmente creamos una regla de puertos para conectarnos con nuestro equipo hacia la VM, para esto seleccionamos la VM y nos vamos al apartado de redes y seleccionamos agregar regla de puerto de entrada. Una vez asignemos el puerto y el protocolo que en nuestro caso fue TCP, guardamos la regla y nos aparecerá en la lista como se muestra en la imagen 15.

The screenshot shows the Microsoft Azure portal interface. The left sidebar is titled 'T2-2019630034-1 | Redes' and includes sections for 'Información general', 'Registro de actividad', 'Control de acceso (IAM)', 'Etiquetas', 'Diagnosticar y solucionar problemas', and 'Configuración' (with 'Redes' selected). The main content area displays the configuration for interface 't2-2019630034-1137'. It shows the IP configuration 'ipconfig1 (Principal)' and the status of the interface. Below this, the 'Reglas de puerto de entrada' (Inbound port rules) section is shown, containing the following table:

Prioridad	Nombre	Puerto	Protocolo	Origen	Destino	Acción
300	SSH	22	TCP	Cualquiera	Cualquiera	<input checked="" type="checkbox"/> Permitir
310	Puerto_5000	5000	TCP	Cualquiera	Cualquiera	<input checked="" type="checkbox"/> Permitir
65000	AllowNetInBound	Cualquier	Cualquier	VirtualNetwork	VirtualNetwork	<input checked="" type="checkbox"/> Permitir
65001	AllowAzureLoadBalancerInBound	Cualquier	Cualquier	AzureLoadBalancer	Cualquiera	<input checked="" type="checkbox"/> Permitir
65500	DenyAllInBound	Cualquier	Cualquier	Cualquiera	Cualquiera	<input type="checkbox"/> Denegar

Imagen 15

Para la máquina T2-2019630034-2 (Nodo 2) y T2-2019630034-3 (Nodo 3) fué el mismo proceso

Conexión con las máquinas virtuales

Una vez se tiene configurado e instalado java en las máquinas virtuales, procedemos a conectarnos a T2-2019630034-1 como se muestra en la imagen 16.

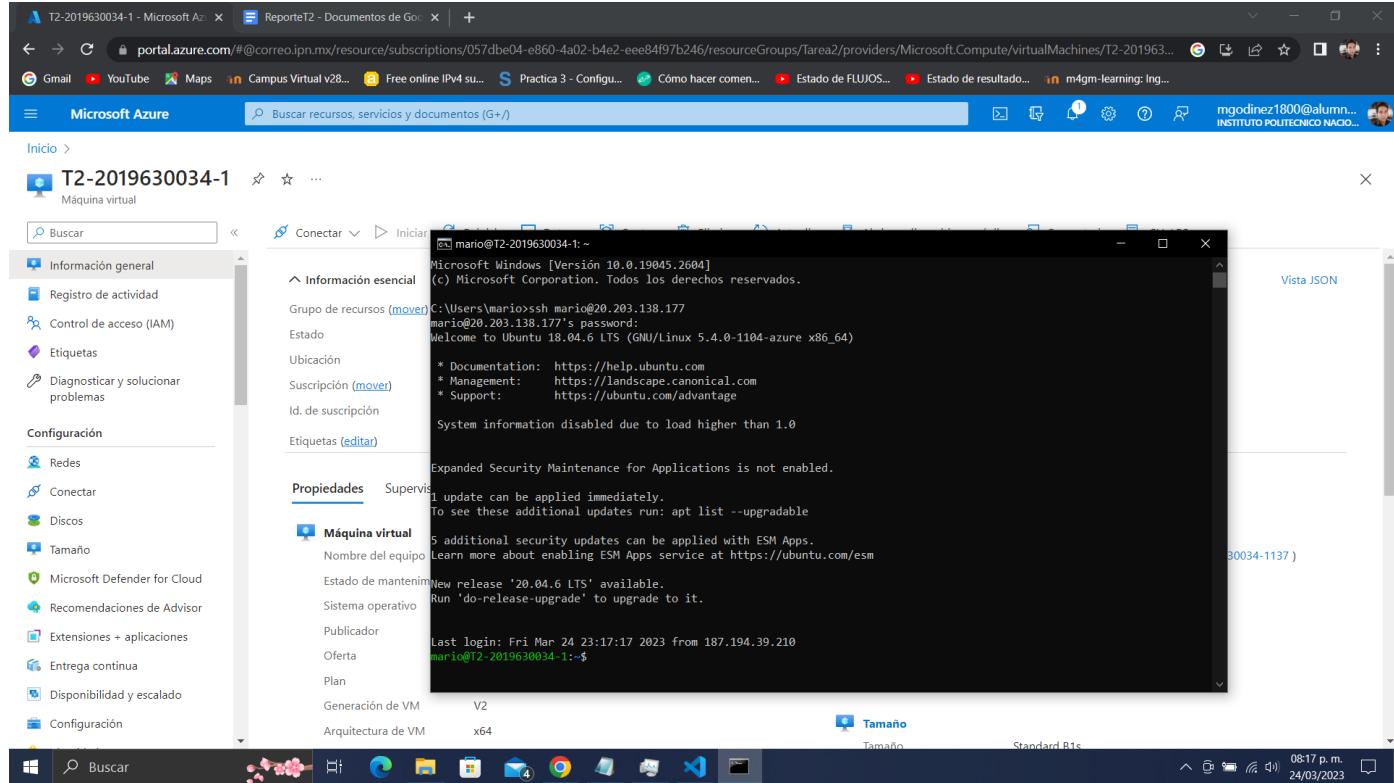


Imagen 16

Lo mismo se hace con el nodo 2 y 3

Carga de programa en las VM

Para cargar el programa se utilizó el protocolo sftp conectándonos con sftp usuario@ip pública como lo muestra la imagen 17.

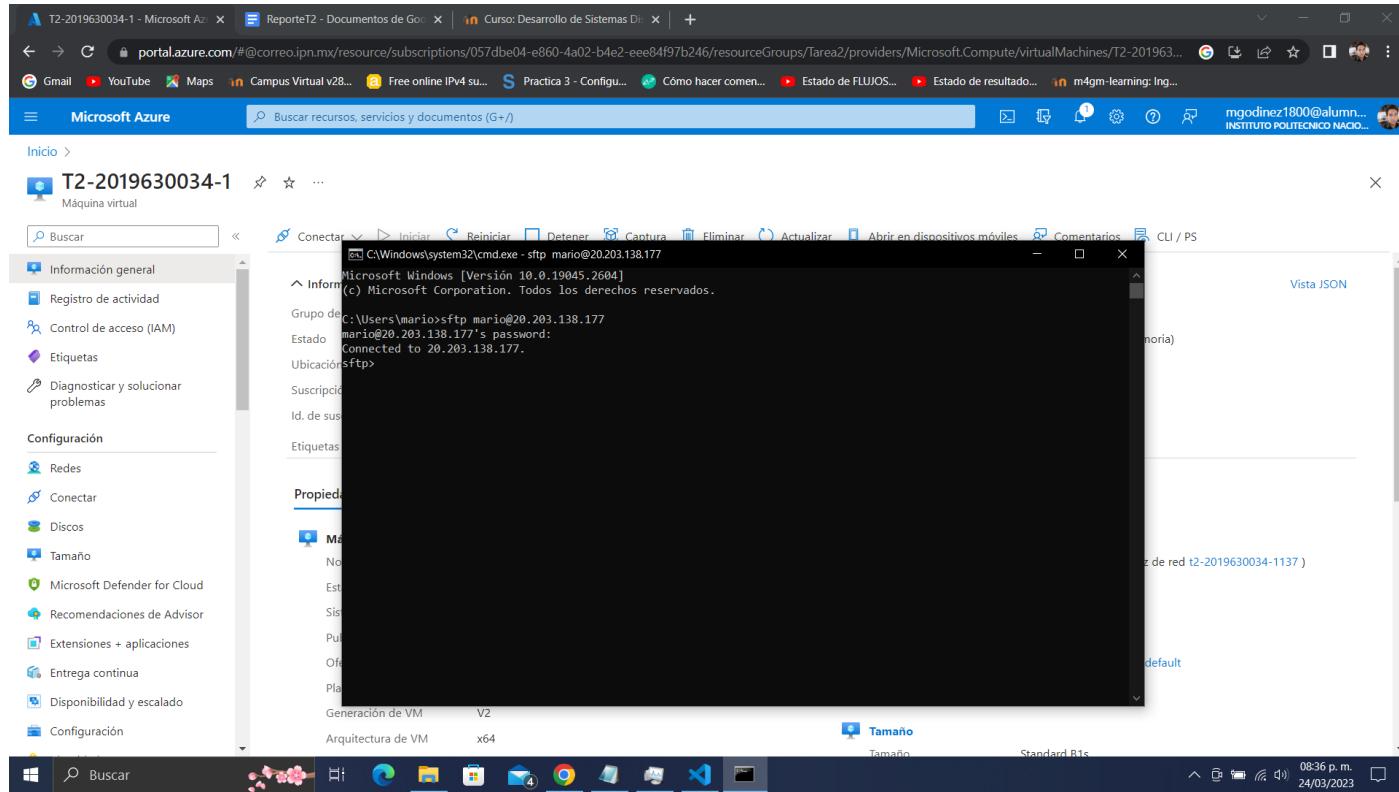
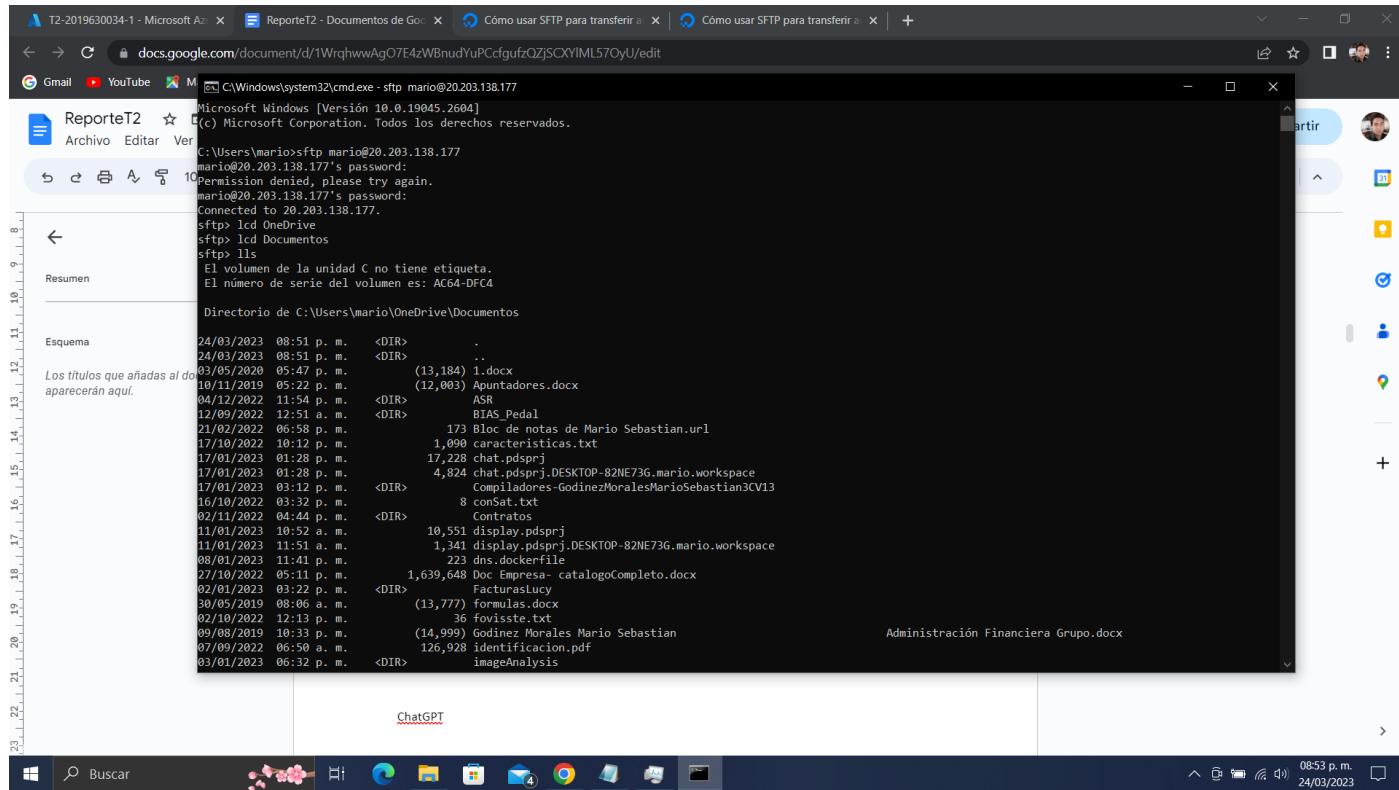


Imagen 17

Posteriormente para navegar en el sistema de archivos utilizamos el comando cd y navegaremos en el sistema de archivos remoto, para navegar en el sistema local se agrega una l al principio del comando como se muestra en la imagen 18.



The screenshot shows a Windows terminal window titled "ReporteT2" with the command "C:\Windows\system32\cmd.exe - sftp mario@20.203.138.177". The output shows:

```
C:\Users\mario>sftp mario@20.203.138.177
mario@20.203.138.177's password:
Permission denied, please try again.
mario@20.203.138.177's password:
Connected to 20.203.138.177.
sftp> lcd OneDrive
sftp> lcd Documentos
sftp> ll
El volumen de la unidad C no tiene etiqueta.
El n&uacute;mero de serie del volumen es: AC64-DFC4

Resumen

Esquema
Los t�tulos que a�adas al do
aparecer�n aqu�.

Directorio de C:\Users\mario\OneDrive\Documentos
24/03/2023 00:51 p. m. <DIR> .
24/03/2023 00:51 p. m. <DIR> ..
03/05/2020 05:47 p. m. (13,184) 1.docx
10/11/2019 05:22 p. m. (12,003) Apuntares.docx
04/12/2022 11:54 p. m. <DIR> ASR
12/09/2022 12:51 a. m. <DIR> BITAS_Pedal
21/02/2022 06:58 p. m. 173 Bloc de notas de Mario Sebastian.url
17/10/2022 10:12 p. m. 1,098 caracteristicas.txt
17/01/2023 01:28 p. m. 17,228 chat.pdspj
17/01/2023 01:28 p. m. 4,824 chat.pdspj.DESKTOP-82NE73G.mario.workspace
17/01/2023 03:12 p. m. <DIR> Compiladores-GodinezMoralesMarioSebastian3CV13
16/10/2022 03:32 p. m. 8 conSat.txt
02/11/2022 04:44 p. m. <DIR> Contratos
11/01/2022 10:52 a. m. 10,551 display.pdspj
11/01/2023 11:51 a. m. 1,341 display.pdspj.DESKTOP-82NE73G.mario.workspace
08/01/2023 11:41 p. m. 223 dns.dockerfile
27/10/2022 05:11 p. m. 1,639,648 Doc Empresa- catalogoCompleto.docx
02/01/2023 03:22 p. m. <DIR> FacturasLucy
30/05/2019 08:06 a. m. (13,777) formulas.docx
02/10/2022 12:13 p. m. 36 fovisste.txt
09/08/2019 10:33 p. m. (14,999) Godinez Morales Mario Sebastian
07/09/2022 06:58 a. m. 126,928 identificacion.pdf
03/01/2023 06:32 p. m. <DIR> imageAnalysis
```

Administraci&on Financiera Grupo.docx

At the bottom of the terminal window, there is a watermark that says "ChatGPT".

Imagen 18

Posteriormente estamos en la ubicación del archivo, utilizamos el comando put seguido del nombre del archivo para enviar el archivo deseado a la VM como se muestra en la imagen 19.

```
C:\Windows\system32\cmd.exe - sftp mario@20.203.138.177
24/03/2023 07:54 p. m. <DIR>      Sistemas Distribuidos
24/03/2023 08:51 p. m. <DIR>      VersionFinal
15/02/2023 06:46 p. m. <DIR>      Visual Studio 2022
25/10/2022 05:45 p. m. <DIR>      Zoom
                           29 archivos   9,542,289 bytes
                           23 dirs   1,783,942,828,032 bytes libres
sftp> lcd VersionFinal
sftp> ll
Tutorials
El volumen de la unidad C no tiene etiqueta.
El n&uacute;mero de serie del volumen es: AC64-DFC4

Directorio de C:\Users\mario\OneDrive\Documentos\VersionFinal

24/03/2023 08:51 p. m. <DIR>      .
24/03/2023 08:51 p. m. <DIR>      421 Nodo0$1.class
24/03/2023 07:48 p. m.           421 Nodo0$2.class
24/03/2023 07:46 p. m.           421 Nodo0$3.class
24/03/2023 07:49 p. m.           4,475 Nodo0.class
24/03/2023 07:49 p. m.           11,567 Nodo0.java
24/03/2023 07:49 p. m.           2,878 Nodo1.class
24/03/2023 07:47 p. m.           4,851 Nodo1.java
24/03/2023 07:50 p. m.           2,878 Nodo2.class
24/03/2023 07:49 p. m.           4,841 Nodo2.java
24/03/2023 07:50 p. m.           2,878 Nodo3.class
24/03/2023 07:50 p. m.           4,850 Nodo3.java
                           11 archivos   40,481 bytes
                           2 dirs   1,783,942,828,032 bytes libres
Cómo transferir archivos con SFTP
sftp> put Nodo1.java
Uploading Nodo1.java to /home/mario/Nodo1.java
Nodo1.java
sftp>
CONTENTS
Cóm o conectarse con SFTP
Cóm o obtener ayuda en SFTP
Cóm o navegar con SFTP
Manipulación de archivos
sftp>
```

This site uses cookies and related technologies, as described in our [privacy policy](#), for purposes that may include site operation, analytics, enhanced user experience, or advertising. You may choose to consent to our use of these technologies, or manage your own preferences.

MANAGE CHOICES

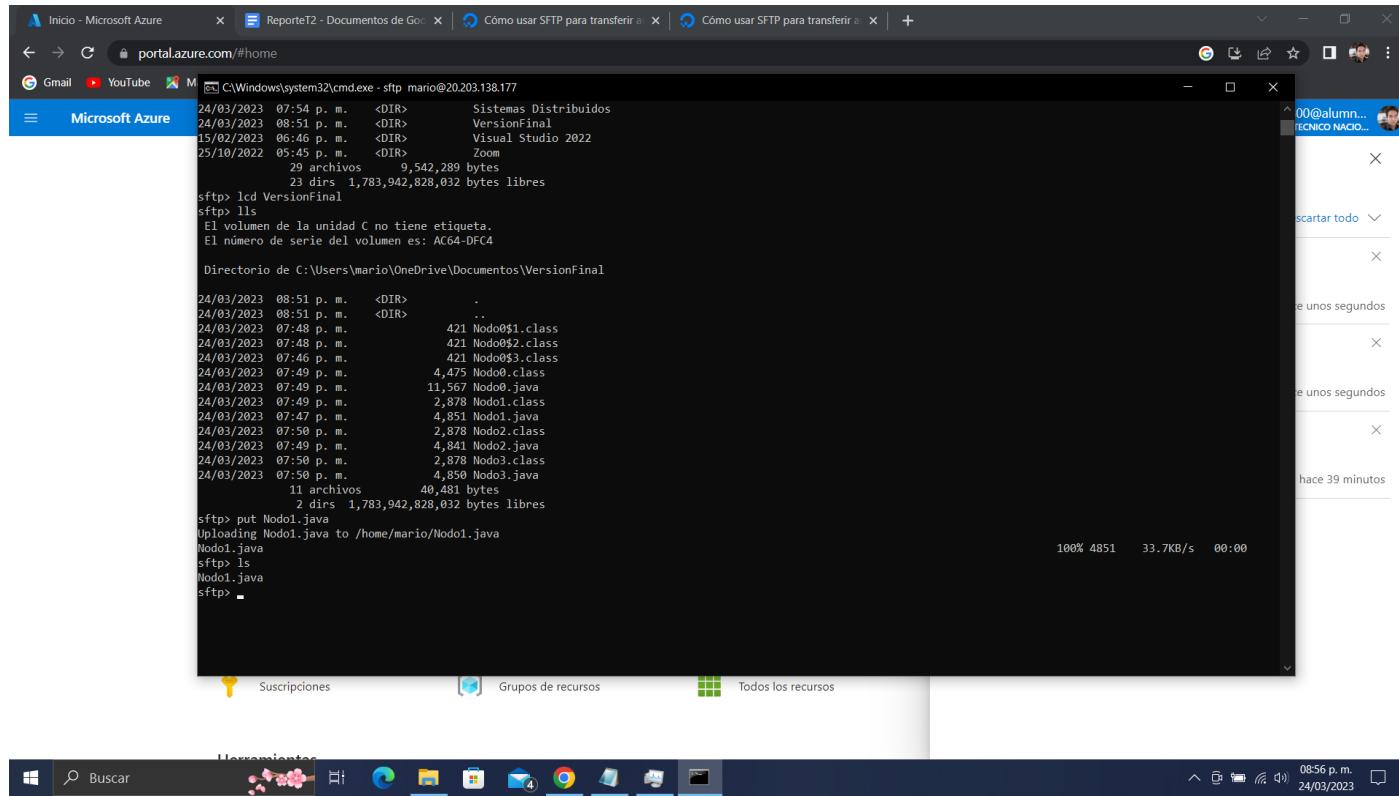
AGREE & PROCEED

100% 4851 33.7KB/s 00:00

08:55 p.m. 24/03/2023

Imagen 19

Verificamos que el archivo se encuentra en la VM con el comando ls como se muestra en la imagen 20.



```
C:\Windows\system32\cmd.exe - sftp mario@20.203.138.177
24/03/2023 07:54 p. m. <DIR> Sistemas Distribuidos
24/03/2023 08:51 p. m. <DIR> VersionFinal
15/02/2023 06:46 p. m. <DIR> Visual Studio 2022
25/10/2022 05:45 p. m. <DIR> Zoom
    29 archivos   9,542,289 bytes
    23 dirs  1,783,942,828,032 bytes libres
sftp> lcd VersionFinal
sftp> ll
El volumen de la unidad C no tiene etiqueta.
El n&uacute;mero de serie del volumen es: AC64-DFC4

Directorio de C:\Users\mario\OneDrive\Documentos\VersionFinal

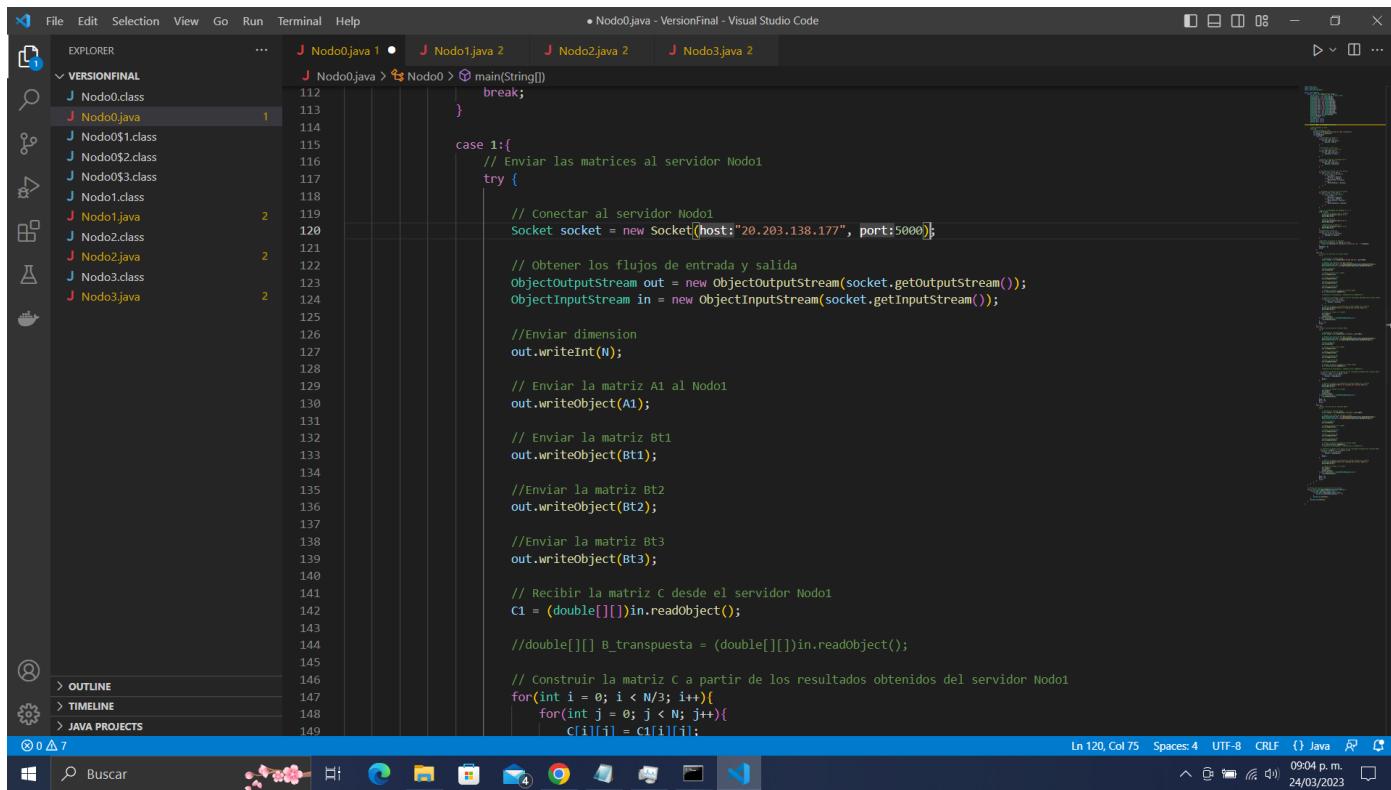
24/03/2023 08:51 p. m. <DIR> .
24/03/2023 08:51 p. m. <DIR> ..
24/03/2023 07:48 p. m. 421 Nodo0$1.class
24/03/2023 07:48 p. m. 421 Nodo0$2.class
24/03/2023 07:46 p. m. 421 Nodo0$3.class
24/03/2023 07:49 p. m. 4,475 Nodo0.class
24/03/2023 07:49 p. m. 11,567 Nodo0.java
24/03/2023 07:49 p. m. 2,878 Nodo1.class
24/03/2023 07:47 p. m. 4,851 Nodo1.java
24/03/2023 07:50 p. m. 2,878 Nodo2.class
24/03/2023 07:49 p. m. 4,841 Nodo2.java
24/03/2023 07:50 p. m. 2,878 Nodo3.class
24/03/2023 07:50 p. m. 4,856 Nodo3.java
    11 archivos   40,481 bytes
    2 dirs  1,783,942,828,032 bytes libres
sftp> put Nodo1.java
Uploading Nodo1.java to /home/mario/Nodo1.java
Nodo1.java
sftp> ls
Nodo1.java
sftp> -
```

Imagen 20

Posteriormente hacemos lo mismo para los otros 2 nodos

Ejecución del programa

Se configura la IP y el puerto en el nodo 0 para que se conecte a los demás nodos como se muestra en la imagen 21.

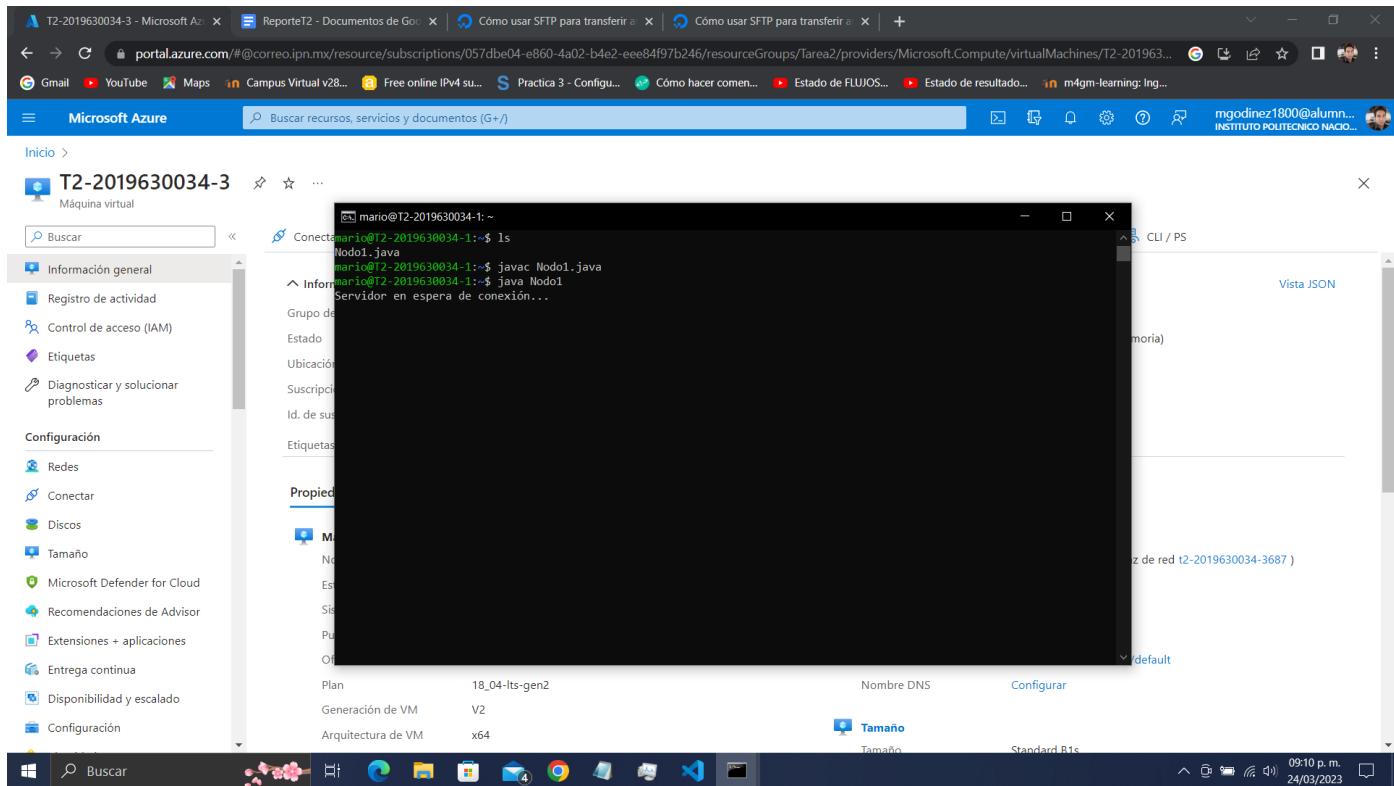


The screenshot shows the Visual Studio Code interface with the following details:

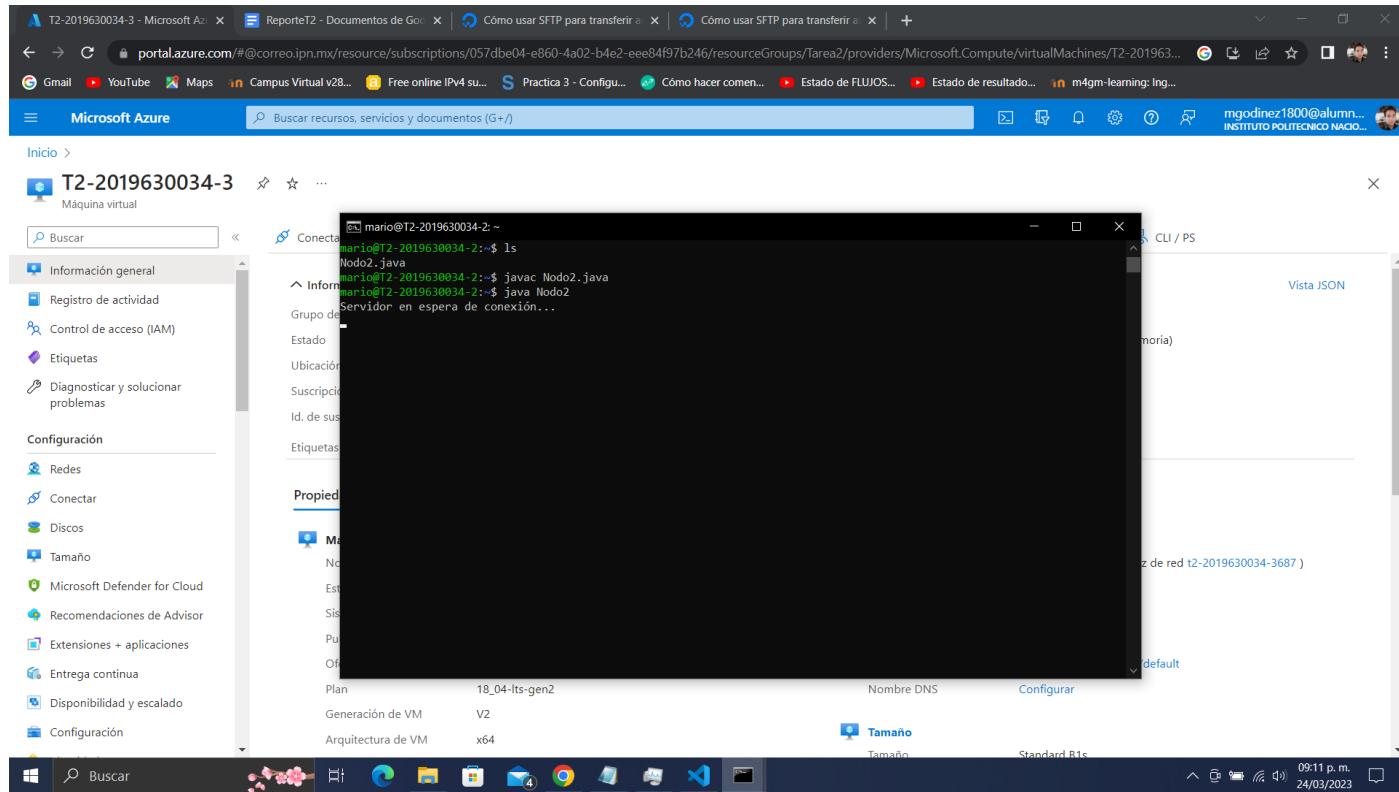
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** • Nodo0.java - VersionFinal - Visual Studio Code.
- Explorer Panel:** Shows a project named "VERSIONFINAL" containing files: Nodo0.class, Nodo0.java (selected), Nodo0\$1.class, Nodo0\$2.class, Nodo0\$3.class, Nodo1.class, Nodo1.java (labeled 1), Nodo2.class, Nodo2.java (labeled 2), Nodo3.class, and Nodo3.java (labeled 2).
- Code Editor:** Displays Java code for the Nodo0.java file. The code handles matrix multiplication between three nodes (Nodo0, Nodo1, and Nodo2). It uses sockets to connect to Nodo1 at port 5000 and Nodo2 at port 5001. It sends matrices A1, Bt1, and Bt2 to Nodo1 and receives matrix C from Nodo1. It then constructs matrix C on Nodo0. Lines 149 and 150 are highlighted.
- Status Bar:** Ln 120, Col 75 | Spaces: 4 | UTF-8 | CRLF | {} Java | 09:04 p.m. | 24/03/2023.

Una vez se tienen configuradas las ip y puertos en donde se va a conectar hacia la VM que en nuestro caso son los puertos 5000, 5001 y 5002 procedemos a correr el código para N = 12.

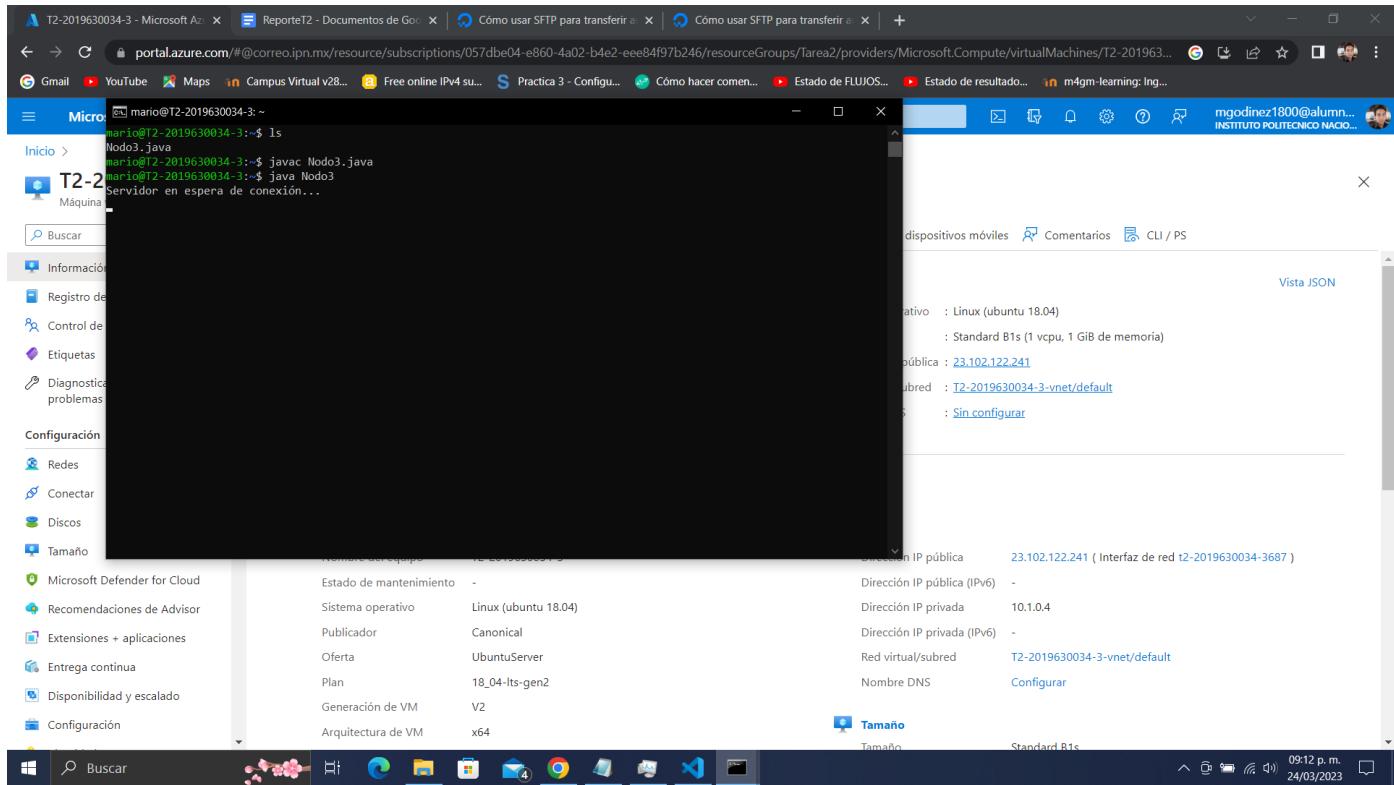
Nodo 1 corriendo:



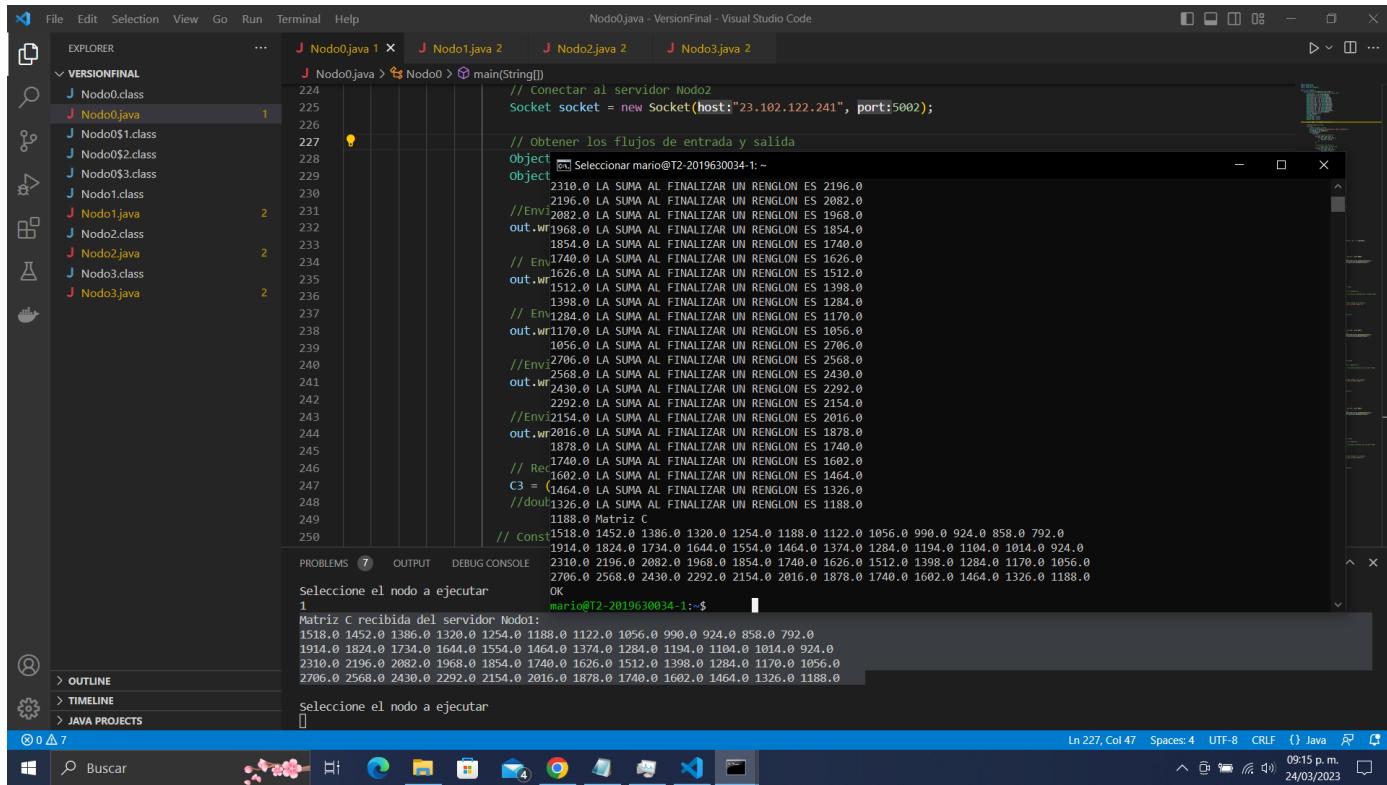
Nodo 2 corriendo:



Nodo 3 corriendo:



Al ejecutar Nodo 0 tenemos:



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files in the 'VERSIONFINAL' folder, including 'Nodo0.java', 'Nodo0.class', 'Nodo0\$1.class', 'Nodo0\$2.class', 'Nodo0\$3.class', 'Nodo1.class', 'Nodo1.java', 'Nodo2.class', 'Nodo2.java', 'Nodo3.class', and 'Nodo3.java'. 'Nodo0.java' is the active file.
- Terminal:** Displays the output of the Java application. The code connects to 'Nodo2' at port 5002, reads input from the server, and prints the received matrix C. The matrix C is a 5x5 grid of floating-point numbers.
- Status Bar:** Shows 'Ln 227, Col 47' and the date '24/03/2023'.

```
// Conectar al servidor Nodo2
Socket socket = new Socket(host:"23.102.122.241", port:5002);

// Obtener los flujos de entrada y salida
Object in = Seleccionar(mario@T2-2019630034-1:-);
Object out = in.getInputStream();

// Envio de datos
out.write("LA SUMA AL FINALIZAR UN RENGLON ES 2196.0
2196.0 LA SUMA AL FINALIZAR UN REnglon ES 2082.0
2082.0 LA SUMA AL FINALIZAR UN RENGLON ES 1968.0
out.write(1968.0 LA SUMA AL FINALIZAR UN RENGLON ES 1854.0
1854.0 LA SUMA AL FINALIZAR UN RENGLON ES 1740.0
// Envio de datos
out.write(1740.0 LA SUMA AL FINALIZAR UN RENGLON ES 1626.0
1626.0 LA SUMA AL FINALIZAR UN RENGLON ES 1512.0
out.write(1512.0 LA SUMA AL FINALIZAR UN RENGLON ES 1398.0
1398.0 LA SUMA AL FINALIZAR UN RENGLON ES 1284.0
// Envio de datos
out.write(1284.0 LA SUMA AL FINALIZAR UN RENGLON ES 1170.0
1170.0 LA SUMA AL FINALIZAR UN RENGLON ES 1056.0
1056.0 LA SUMA AL FINALIZAR UN RENGLON ES 2706.0
// Envio de datos
2706.0 LA SUMA AL FINALIZAR UN RENGLON ES 2568.0
out.write(2568.0 LA SUMA AL FINALIZAR UN RENGLON ES 2430.0
2430.0 LA SUMA AL FINALIZAR UN RENGLON ES 2292.0
2292.0 LA SUMA AL FINALIZAR UN RENGLON ES 2154.0
// Envio de datos
2154.0 LA SUMA AL FINALIZAR UN RENGLON ES 2016.0
out.write(2016.0 LA SUMA AL FINALIZAR UN RENGLON ES 1878.0
1878.0 LA SUMA AL FINALIZAR UN RENGLON ES 1740.0
1740.0 LA SUMA AL FINALIZAR UN RENGLON ES 1602.0
// Recibo datos
1602.0 LA SUMA AL FINALIZAR UN RENGLON ES 1464.0
C3 = (1464.0 LA SUMA AL FINALIZAR UN RENGLON ES 1326.0
//dout
1326.0 LA SUMA AL FINALIZAR UN RENGLON ES 1188.0
1188.0 Matriz C
// Const
1914.0 1824.0 1734.0 1644.0 1554.0 1464.0 1374.0 1284.0 1194.0 1104.0 1014.0 924.0
1914.0 1824.0 1734.0 1644.0 1554.0 1464.0 1374.0 1284.0 1194.0 1104.0 1014.0 924.0
1518.0 1452.0 1386.0 1320.0 1254.0 1188.0 1122.0 1056.0 990.0 924.0 858.0 792.0
1914.0 1824.0 1734.0 1644.0 1554.0 1464.0 1374.0 1284.0 1194.0 1104.0 1014.0 924.0
2310.0 2196.0 2082.0 1968.0 1854.0 1740.0 1626.0 1512.0 1398.0 1284.0 1170.0 1056.0
2706.0 2568.0 2430.0 2292.0 2154.0 2016.0 1878.0 1740.0 1602.0 1464.0 1326.0 1188.0
Matriz C recibida del servidor Nodo1:
1518.0 1452.0 1386.0 1320.0 1254.0 1188.0 1122.0 1056.0 990.0 924.0 858.0 792.0
1914.0 1824.0 1734.0 1644.0 1554.0 1464.0 1374.0 1284.0 1194.0 1104.0 1014.0 924.0
2310.0 2196.0 2082.0 1968.0 1854.0 1740.0 1626.0 1512.0 1398.0 1284.0 1170.0 1056.0
2706.0 2568.0 2430.0 2292.0 2154.0 2016.0 1878.0 1740.0 1602.0 1464.0 1326.0 1188.0
```

En la terminal de VS code de manera local se ejecutó el nodo 0 y posteriormente se mandaron los datos al nodo 1 que es la VM en la terminal CMD

Al ejecutar Nodo 2 tenemos:

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files in the project: `Nodo0.java`, `Nodo1.java`, `Nodo2.java`, and `Nodo3.java`.
- Code Editor:** The `Nodo2.java` file is open, showing Java code for a client node. It uses sockets to communicate with a server node (`Nodo0`). The code performs matrix operations (addition and multiplication) on matrices received from other nodes.
- Terminal:** The terminal window shows the output of the execution. It includes the command run, the host IP and port, and the resulting matrix `C3` received from the server.
- Status Bar:** Shows the current line (Ln 227), column (Col 47), and other system information like date and time (24/03/2023, 09:19 p.m.).

```
// Conectar al servidor Nodo2
Socket socket = new Socket(host:"23.102.122.241", port:5002);

// Obtener los flujos de entrada y salida
ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());
ObjectInputStream in = new ObjectInputStream(socket.getInputStream());

// Enviar dimension
out.writeInt(N);

// Enviar la matriz A2 al Nodo2
out.writeObject(A2);

// Enviar la matriz Bt1
out.writeObject(Bt1);

// Enviar la matriz Bt2
out.writeObject(Bt2);

// Enviar la matriz Bt3
out.writeObject(Bt3);

// Recibir la matriz C3 desde el nodo 0
C3 = (double[][])(in.readObject());
B_transpuesta = (double[][])(in.readObject());
//double[][] B_transpuesta = (double[][])(in.readObject());

// Construir la matriz C3 a partir de las matrices recibidas
for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++) {
        C3[i][j] = 0;
        for (int k = 0; k < N; k++) {
            C3[i][j] += A2[i][k] * B_transpuesta[k][j];
        }
    }
}

// Imprimir la matriz C3
for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++) {
        System.out.print(C3[i][j] + " ");
    }
    System.out.println();
}
```

```
mario@T2-2019630034-2:~$
```

```
Selecione el nodo a ejecutar
2
Matriz C2 recibida del servidor Nodo2:
3102.0 2940.0 2778.0 2616.0 2454.0 2292.0 2130.0 1968.0 1806.0 1644.0 1482.0 1320.0
3498.0 3312.0 3126.0 2940.0 2754.0 2568.0 2382.0 2196.0 2010.0 1824.0 1638.0 1452.0
3894.0 3684.0 3474.0 3264.0 3054.0 2844.0 2634.0 2424.0 2214.0 2004.0 1794.0 1584.0
4290.0 4056.0 3822.0 3588.0 3354.0 3120.0 2886.0 2652.0 2418.0 2184.0 1950.0 1716.0
```

Al ejecutar Nodo 3 tenemos:

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project named "VERSIONFINAL" containing files: Nodo0.class, Nodo0\$1.class, Nodo0\$2.class, Nodo0\$3.class, Nodo1.class, Nodo1.java (selected), Nodo2.class, Nodo2.java, Nodo3.class, and Nodo3.java.
- Terminal:** Displays the output of the Java application execution. The code connects to Nodo0 at port 5002, sends matrix dimensions, and receives matrix C3 from the server. It then constructs matrix C by summing matrices A and B.
- Output:** Shows the final matrix C3 received from the server.
- Status Bar:** Includes information like line (Ln 227), column (Col 47), spaces (Spaces: 4), encoding (UTF-8), and date/time (24/03/2023 09:20 p.m.).

```
// Conectar al servidor Nodo0
Socket socket = new Socket(host:"23.102.122.241", port:5002);

// Obtener los flujos de entrada y salida
ObjectOutputStream mario@T2-2019630034-3:~>
ObjectInputStream
5478,0 LA SUMA AL FINALIZAR UN RENGLON ES 5172,0
5172,0 LA SUMA AL FINALIZAR UN RENGLON ES 4866,0
//Enviar dimension 4866,0 LA SUMA AL FINALIZAR UN RENGLON ES 4560,0
out.writeInt(4);
4560,0 LA SUMA AL FINALIZAR UN RENGLON ES 4254,0
4254,0 LA SUMA AL FINALIZAR UN RENGLON ES 3948,0
// Enviar la matriz 3948,0 LA SUMA AL FINALIZAR UN RENGLON ES 3642,0
3642,0 LA SUMA AL FINALIZAR UN RENGLON ES 3336,0
out.writeObject(A);
3336,0 LA SUMA AL FINALIZAR UN RENGLON ES 3030,0
3030,0 LA SUMA AL FINALIZAR UN RENGLON ES 2724,0
// Enviar la matriz 2724,0 LA SUMA AL FINALIZAR UN RENGLON ES 2418,0
2418,0 LA SUMA AL FINALIZAR UN RENGLON ES 2112,0
2112,0 LA SUMA AL FINALIZAR UN RENGLON ES 5874,0
//Enviar la matriz 5874,0 LA SUMA AL FINALIZAR UN RENGLON ES 5544,0
5544,0 LA SUMA AL FINALIZAR UN RENGLON ES 5214,0
out.writeObject(B);
5214,0 LA SUMA AL FINALIZAR UN RENGLON ES 4884,0
4884,0 LA SUMA AL FINALIZAR UN RENGLON ES 4554,0
//Enviar la matriz 4554,0 LA SUMA AL FINALIZAR UN RENGLON ES 4224,0
4224,0 LA SUMA AL FINALIZAR UN RENGLON ES 3894,0
3894,0 LA SUMA AL FINALIZAR UN RENGLON ES 3564,0
3564,0 LA SUMA AL FINALIZAR UN RENGLON ES 3234,0
// Recibir la matriz 3234,0 LA SUMA AL FINALIZAR UN RENGLON ES 2904,0
C3 = (double[][]);i
2904,0 LA SUMA AL FINALIZAR UN RENGLON ES 2574,0
//double[][] B_traspose;
2574,0 LA SUMA AL FINALIZAR UN RENGLON ES 2244,0
2244,0 Matriz C
4686,0 4428,0 4170,0 3912,0 3654,0 3396,0 3138,0 2880,0 2622,0 2364,0 2106,0 1848,0
5082,0 4800,0 4518,0 4236,0 3954,0 3672,0 3390,0 3108,0 2826,0 2544,0 2262,0 1980,0
5478,0 5172,0 4866,0 4560,0 4254,0 3948,0 3642,0 3336,0 3030,0 2724,0 2418,0 2112,0
5874,0 5544,0 5214,0 4884,0 4554,0 4224,0 3894,0 3564,0 3234,0 2904,0 2574,0 2244,0
// Construir la matriz
5082,0 4800,0 4518,0 4236,0 3954,0 3672,0 3390,0 3108,0 2826,0 2544,0 2262,0 1980,0
5478,0 5172,0 4866,0 4560,0 4254,0 3948,0 3642,0 3336,0 3030,0 2724,0 2418,0 2112,0
5874,0 5544,0 5214,0 4884,0 4554,0 4224,0 3894,0 3564,0 3234,0 2904,0 2574,0 2244,0
OK
mario@T2-2019630034-3:~$
```

Selección del nodo a ejecutar
3

Selección del nodo a ejecutar
[]

Finalmente las matrices A, B son:

```
// Conectar al servidor Nodo2
Socket socket = new Socket(host:"23.102.122.241", port:5002);

// Obtener los flujos de entrada y salida
ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());
ObjectInputStream in = new ObjectInputStream(socket.getInputStream());

//Enviar dimension
out.writeInt(N);
```

PS C:\Users\mario\OneDrive\Documentos\Sistemas Distribuidos\VersionFinal> cd "c:\Users\mario\OneDrive\Documentos\Sistemas Distribuidos\VersionFinal" ; if (\$?) { java ac Nodo0.java } ; if (\$?) { java Nodo0 }

Seleccione el nodo a ejecutar

0

Matriz A:

0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0	10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0
2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0	12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0
4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0	14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0 25.0
6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0	16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0 25.0 26.0 27.0
8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0	18.0 19.0 20.0 21.0 22.0 23.0 24.0 25.0 26.0 27.0 28.0 29.0
10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0	20.0 21.0 22.0 23.0 24.0 25.0 26.0 27.0 28.0 29.0 30.0 31.0
12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0	22.0 23.0 24.0 25.0 26.0 27.0 28.0 29.0 30.0 31.0 32.0 33.0

Matriz B:

0.0 3.0 6.0 9.0 12.0 15.0 18.0 21.0 24.0 27.0 30.0 33.0	-1.0 2.0 5.0 8.0 11.0 14.0 17.0 20.0 23.0 26.0 29.0 32.0
-2.0 1.0 4.0 7.0 10.0 13.0 16.0 19.0 22.0 25.0 28.0 31.0	-3.0 0.0 3.0 6.0 9.0 12.0 15.0 18.0 21.0 24.0 27.0 30.0
-4.0 -1.0 2.0 5.0 8.0 11.0 14.0 17.0 20.0 23.0 26.0 29.0	-5.0 -2.0 1.0 4.0 7.0 10.0 13.0 16.0 19.0 22.0 25.0 28.0
-6.0 -3.0 0.0 3.0 6.0 9.0 12.0 15.0 18.0 21.0 24.0 27.0	-7.0 -4.0 -1.0 2.0 5.0 8.0 11.0 14.0 17.0 20.0 23.0 26.0
-8.0 -5.0 -2.0 1.0 4.0 7.0 10.0 13.0 16.0 19.0 22.0 25.0	-9.0 -6.0 -3.0 0.0 3.0 6.0 9.0 12.0 15.0 18.0 21.0 24.0
-10.0 -7.0 -4.0 -1.0 2.0 5.0 8.0 11.0 14.0 17.0 20.0 23.0	-11.0 -8.0 -5.0 -2.0 1.0 4.0 7.0 10.0 13.0 16.0 19.0 22.0

Matriz C y Checksum:

```
// Conectar al servidor Nodo2
Socket socket = new Socket(host:"23.102.122.241", port:5002);

// Obtener los flujos de entrada y salida
ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());
ObjectInputStream in = new ObjectInputStream(socket.getInputStream());

//Enviar dimension
out.writeInt(N);
```

Matriz C:

1518.0 1452.0 1386.0 1320.0 1254.0 1188.0 1122.0 1056.0 998.0 924.0 858.0 792.0	1914.0 1824.0 1734.0 1644.0 1554.0 1464.0 1374.0 1284.0 1194.0 1104.0 1014.0 924.0
2310.0 2196.0 2082.0 1968.0 1854.0 1740.0 1626.0 1512.0 1398.0 1284.0 1170.0 1056.0	2706.0 2568.0 2430.0 2292.0 2154.0 2016.0 1878.0 1740.0 1692.0 1464.0 1326.0 1188.0
3102.0 2940.0 2778.0 2616.0 2454.0 2292.0 2130.0 1968.0 1886.0 1644.0 1482.0 1320.0	3498.0 3312.0 3126.0 2940.0 2754.0 2568.0 2382.0 2196.0 2010.0 1824.0 1638.0 1452.0
3894.0 3684.0 3474.0 3264.0 3054.0 2844.0 2634.0 2424.0 2214.0 2004.0 1794.0 1584.0	4290.0 4056.0 3822.0 3588.0 3354.0 3120.0 2886.0 2652.0 2418.0 2184.0 1950.0 1716.0
4686.0 4428.0 4179.0 3912.0 3654.0 3396.0 3138.0 2880.0 2622.0 2364.0 2106.0 1848.0	5082.0 4800.0 4518.0 4236.0 3954.0 3672.0 3390.0 3108.0 2826.0 2544.0 2262.0 1980.0
5478.0 5172.0 4866.0 4560.0 4254.0 3948.0 3642.0 3336.0 3030.0 2724.0 2418.0 2112.0	5874.0 5544.0 5214.0 4884.0 4554.0 4224.0 3894.0 3564.0 3234.0 2904.0 2574.0 2244.0

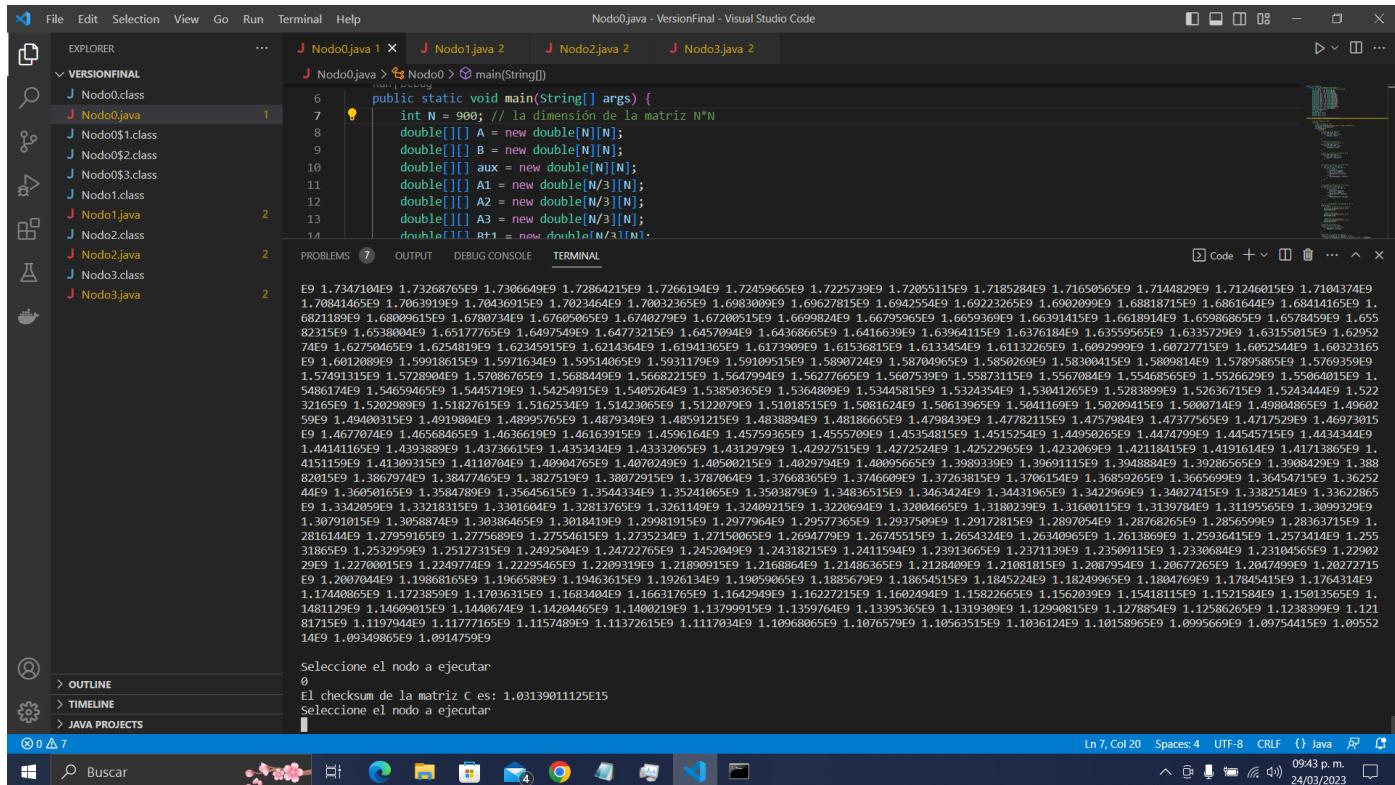
El checksum de la matriz C es: 375408.0

Seleccione el nodo a ejecutar

Checksum = 375408.0

Para N = 900

Al ejecutar Nodo 1 2 y 3 tenemos como checksum:



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files: `Nodo0.class`, `Nodo$1.class`, `Nodo$2.class`, `Nodo$3.class`, `Nodo1.class`, `Nodo1.java`, `Nodo2.class`, `Nodo2.java`, `Nodo3.class`, and `Nodo3.java`. `Nodo3.java` is the active file.
- Code Editor:** Displays the `Nodo3.java` code, which contains a main method for matrix multiplication. The variable `N` is set to 900. The code uses nested loops to calculate the product of two 900x900 matrices, `A` and `B`.
- Terminal:** Shows the command `javac Nodo3.java` followed by the output of the compilation process.
- Status Bar:** Shows the current file is `Nodo3.java`, the line number is 1, the column is 7, and the character is a space. It also shows the date and time as 24/03/2023 09:43 p.m.

checksum = 1.03139011125E15

Finalmente para N = 3000

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project named "TAREA2.9.1" containing several Java files: Nodo0.class, Nodo0\$1.class, Nodo0\$2.class, Nodo0\$3.class, Nodo1.class, Nodo2.class, and Nodo3.class. The file "Nodo0.java" is currently selected.
- Code Editor:** Displays the content of "Nodo0.java". The code initializes four 3x3 double arrays (A, B, A1, A2) and three 3x3 double arrays (aux, Bt1, Bt2, Bt3, C1, C2, C3). It uses a Scanner object to read input from System.in. A comment indicates that the dimension of the matrix is N*N. The code then performs matrix multiplication and calculates a checksum.
- Terminal:** Shows the output of the program execution. The output includes the message "OK", the instruction "Seleccione el nodo a ejecutar", the value "0", and the message "El checksum de la matriz C es: 4.2500703376504915E17".
- Status Bar:** Shows the current line (Ln 7, Col 21), spaces (Spaces: 4), encoding (UTF-8), and file type (CRLF). It also displays the date and time (24/03/2023, 11:44 p.m.).

Checksum = 4.2500703376504915E17

Conversación con chatGPT

Para empezar con el programa, primero lo modulé en programas más sencillos, para empezar le pedí a chatGPT que hiciera un programa en java que inicializará 2 matrices conforme a las reglas dadas en la tarea. La conversación es la siguiente:

Mario Sebastian

Hola

ChatGPT

¡Hola! ¿En qué puedo ayudarte hoy?

Mario Sebastian

Por favor crea un programa en java que inicialice 2 matrices A y B, en donde A[i][j] y B[i][j] son elementos de las matrices A y B respectivamente y A[i][j] = 2*i+j , B[i][j] = 3*i-j, la dimensión de la matriz es N*N.

ChatGPT

¡Claro! Aquí te muestro cómo crear el programa en Java que inicializa dos matrices A y B, según las condiciones que indicaste:

```
public class Matrices {  
    public static void main(String[] args) {  
        int N = 4; // la dimensión de la matriz N*N  
        int[][] A = new int[N][N];  
        int[][] B = new int[N][N];  
  
        // Inicializar la matriz A  
        for(int i=0; i<N; i++) {  
            for(int j=0; j<N; j++) {  
                A[i][j] = 2*i+j;  
            }  
        }  
  
        // Inicializar la matriz B  
        for(int i=0; i<N; i++) {  
            for(int j=0; j<N; j++) {  
                B[i][j] = 3*i-j;  
            }  
        }  
  
        // Mostrar las matrices A y B en la consola  
        System.out.println("Matriz A:");  
        mostrarMatriz(A);
```

```

System.out.println("Matriz B:");
mostrarMatriz(B);
}

// Método para mostrar una matriz en la consola
public static void mostrarMatriz(int[][] matriz) {
    for(int i=0; i<matriz.length; i++) {
        for(int j=0; j<matriz[0].length; j++) {
            System.out.print(matriz[i][j] + " ");
        }
        System.out.println();
    }
    System.out.println();
}
}

```

En este programa, creamos dos matrices A y B de tamaño N*N, luego las inicializamos con los valores correspondientes utilizando dos bucles for anidados, y finalmente las mostramos en la consola utilizando el método mostrarMatriz(). En la salida de consola, primero se mostrará la matriz A y luego la matriz B. Puedes cambiar el valor de N para ajustar la dimensión de la matriz.

Mario Sebastian

ahora modifica el programa para que adicionalmente se calcule la matriz transpuesta de B

ChatGPT

¡Claro! Aquí te muestro cómo modificar el programa anterior para calcular la matriz transpuesta de B:

```

public class Matrices {
    public static void main(String[] args) {

```

```
int N = 4; // la dimensión de la matriz N*N

int[][] A = new int[N][N];

int[][] B = new int[N][N];

int[][] Bt = new int[N][N]; // matriz transpuesta de B

// Inicializar la matriz A

for(int i=0; i<N; i++) {

    for(int j=0; j<N; j++) {

        A[i][j] = 2*i+j;

    }

}

// Inicializar la matriz B

for(int i=0; i<N; i++) {

    for(int j=0; j<N; j++) {

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

    }

}

// Calcular la matriz transpuesta de B

for(int i=0; i<N; i++) {

    for(int j=0; j<N; j++) {

        Bt[i][j] = B[j][i];

    }

}

// Mostrar las matrices A, B y Bt en la consola

System.out.println("Matriz A:");

mostrarMatriz(A);
```

```

System.out.println("Matriz B:");
mostrarMatriz(B);

System.out.println("Matriz transpuesta de B:");
mostrarMatriz(Bt);

}

// Método para mostrar una matriz en la consola
public static void mostrarMatriz(int[][] matriz) {
    for(int i=0; i<matriz.length; i++) {
        for(int j=0; j<matriz[0].length; j++) {
            System.out.print(matriz[i][j] + " ");
        }
        System.out.println();
    }
    System.out.println();
}

```

En este programa, hemos agregado una nueva matriz Bt para almacenar la matriz transpuesta de B. Luego, hemos utilizado dos bucles for anidados para calcular los elementos de Bt y los hemos mostrado en la consola después de la matriz B. La salida de consola mostrará la matriz A, B y la matriz transpuesta de B.

Mario Sebastian

modifica el código para que la matriz transpuesta se almacene en la matriz B

ChatGPT

¡Por supuesto! Aquí te muestro cómo modificar el programa anterior para almacenar la matriz transpuesta en la matriz B:

```
public class Matrices {  
  
    public static void main(String[] args) {  
  
        int N = 4; // la dimensión de la matriz N*N  
  
        int[][] A = new int[N][N];  
  
        int[][] B = new int[N][N];  
  
        // Inicializar la matriz A  
  
        for(int i=0; i<N; i++) {  
  
            for(int j=0; j<N; j++) {  
  
                A[i][j] = 2*i+j;  
  
            }  
  
        }  
  
        // Inicializar la matriz B y calcular su transpuesta  
  
        for(int i=0; i<N; i++) {  
  
            for(int j=0; j<N; j++) {  
  
                B[i][j] = 3*i-j;  
  
            }  
  
        }  
  
        // Calcular la matriz transpuesta de B mientras se inicializa  
  
        B[j][i] = 3*i-j;  
  
    }  
  
    // Mostrar las matrices A y B en la consola  
  
    System.out.println("Matriz A:");  
  
    mostrarMatriz(A);
```

```

        System.out.println("Matriz B:");
        mostrarMatriz(B);
    }

    // Método para mostrar una matriz en la consola
    public static void mostrarMatriz(int[][] matriz) {
        for(int i=0; i<matriz.length; i++) {
            for(int j=0; j<matriz[0].length; j++) {
                System.out.print(matriz[i][j] + " ");
            }
            System.out.println();
        }
        System.out.println();
    }
}

```

Mario Sebastian

modifica el código para que especifiques que estas imprimiendo la matriz transpuesta y no B

ChatGPT

¡Claro! Aquí te muestro cómo modificar el programa anterior para que se especifique que se está imprimiendo la matriz transpuesta:

```

public class Matrices {

    public static void main(String[] args) {
        int N = 4; // la dimensión de la matriz N*N
        int[][] A = new int[N][N];
        int[][] B = new int[N][N];

        // Inicializar la matriz A

```

```

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

// Inicializar la matriz B y calcular su transpuesta
for(int i=0; i<N; i++) {
    for(int j=0; j<N; j++) {
        B[i][j] = 3*i-j;
    }
}

// Calcular la matriz transpuesta de B mientras se inicializa
B[j][i] = 3*i-j;
}

}

// Mostrar la matriz A en la consola
System.out.println("Matriz A:");
mostrarMatriz(A);

// Mostrar la matriz transpuesta de B en la consola
System.out.println("Matriz transpuesta de B:");
mostrarMatriz(B);
}

}

// Método para mostrar una matriz en la consola
public static void mostrarMatriz(int[][] matriz) {
    for(int i=0; i<matriz.length; i++) {
        for(int j=0; j<matriz[0].length; j++) {

```

```

        System.out.print(matriz[i][j] + " ");
    }

    System.out.println();
}

System.out.println();
}

}

```

En este programa, hemos agregado una nueva línea de código antes de imprimir la matriz transpuesta que especifica que se está imprimiendo la matriz transpuesta de B. Luego, hemos mostrado la matriz transpuesta de B en la consola. La salida de consola mostrará la matriz A y la matriz B transpuesta, pero ahora se especifica claramente qué matriz se está imprimiendo.

Una vez teníamos la primera parte que consiste en inicializar las matrices A, B y obtener la transpuesta de B, el siguiente paso es mandarlas por un socket TCP, por que le pedí a chatGPT en una nueva conversación que modificará el código para que las matrices se mandaran por sockets de la siguiente manera:

Mario Sebastian

el siguiente código de java inicializa 2 matrices A, B y calcula la matriz transpuesta de B, modifica el código para que pueda mandar las matrices por un socket TCP hacia un servidor: public class Matrices {

```

public static void main(String[] args) {

    int N = 4; // la dimensión de la matriz N*N

    int[][] A = new int[N][N];
    int[][] B = new int[N][N];

    // Inicializar la matriz A

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

```

```

// Inicializar la matriz B y calcular su transpuesta
for(int i=0; i<N; i++) {
    for(int j=0; j<N; j++) {
        B[i][j] = 3*i-j;

        // Calcular la matriz transpuesta de B mientras se inicializa
        B[j][i] = 3*i-j;
    }
}

// Mostrar la matriz A en la consola
System.out.println("Matriz A:");
mostrarMatriz(A);

// Mostrar la matriz transpuesta de B en la consola
System.out.println("Matriz transpuesta de B:");
mostrarMatriz(B);

}

// Método para mostrar una matriz en la consola
public static void mostrarMatriz(int[][] matriz) {
    for(int i=0; i<matriz.length; i++) {
        for(int j=0; j<matriz[0].length; j++) {
            System.out.print(matriz[i][j] + " ");
        }
        System.out.println();
    }
    System.out.println();
}

```

```
 }  
 }
```

ChatGPT

Para enviar las matrices a través de un socket TCP hacia un servidor, primero necesitarás convertirlas en un formato que se pueda enviar a través del socket. Una opción sería convertir cada matriz en una cadena de caracteres y enviar esas cadenas al servidor. Luego, el servidor podría convertir las cadenas de caracteres de nuevo en matrices.

A continuación se muestra cómo modificar el código para enviar las matrices por un socket TCP:

```
import java.io.*;  
import java.net.*;  
  
public class Matrices {  
    public static void main(String[] args) {  
        int N = 4; // la dimensión de la matriz N*N  
        int[][] A = new int[N][N];  
        int[][] B = new int[N][N];  
  
        // Inicializar la matriz A  
        for(int i=0; i<N; i++) {  
            for(int j=0; j<N; j++) {  
                A[i][j] = 2*i+j;  
            }  
        }  
  
        // Inicializar la matriz B y calcular su transpuesta  
        for(int i=0; i<N; i++) {  
            for(int j=0; j<N; j++) {  
                B[i][j] = 3*i-j;  
            }  
        }
```

```
// Calcular la matriz transpuesta de B mientras se inicializa  
  
    B[j][i] = 3*i-j;  
}  
  
}  
  
  
// Mostrar la matriz A en la consola  
  
System.out.println("Matriz A:");  
mostrarMatriz(A);  
  
  
// Mostrar la matriz transpuesta de B en la consola  
  
System.out.println("Matriz transpuesta de B:");  
mostrarMatriz(B);  
  
  
// Enviar las matrices al servidor  
  
try {  
    // Conectar al servidor  
  
    Socket socket = new Socket("direccion_del_servidor", puerto_del_servidor);  
  
  
    // Obtener los flujos de entrada y salida  
  
    ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());  
    ObjectInputStream in = new ObjectInputStream(socket.getInputStream());  
  
  
    // Enviar la matriz A  
  
    out.writeObject(A);  
  
  
    // Enviar la matriz B  
  
    out.writeObject(B);
```

```

// Recibir la matriz transpuesta de B desde el servidor
int[][] B_transpuesta = (int[][])(in.readObject());

// Mostrar la matriz transpuesta de B recibida del servidor en la consola
System.out.println("Matriz transpuesta de B recibida del servidor:");
mostrarMatriz(B_transpuesta);

// Cerrar los flujos y el socket
out.close();
in.close();
socket.close();

} catch (IOException | ClassNotFoundException e) {
    e.printStackTrace();
}

}

// Método para mostrar una matriz en la consola
public static void mostrarMatriz(int[][] matriz) {
    for(int i=0; i<matriz.length; i++) {
        for(int j=0; j<matriz[0].length; j++) {
            System.out.print(matriz[i][j] + " ");
        }
        System.out.println();
    }
    System.out.println();
}
}

```

En este código, se usa un ObjectOutputStream para enviar las matrices al servidor y un ObjectInputStream para recibir la matriz transpuesta de B del servidor. También se proporciona la

dirección del servidor y el puerto del servidor en el que se escuchará la conexión del socket. Asegúrate de modificar estos valores para que coincidan con la configuración del servidor.

Hasta este punto ya tenemos la inicialización de matrices y el cliente que manda las matrices, ahora teníamos el problema de los servidores, en otra conversación le pedí a chatGPT que programara un servidor de la siguiente manera:

Mario Sebastian

programa en java un servidor que este en espera de una conexión por un cliente, dicho servidor recibirá 2 matrices A y B de dimensión N y mostrará en consola dichas matrices

ChatGPT

Aquí te proporciono un ejemplo de código en Java para un servidor que espera la conexión de un cliente y recibe dos matrices A y B de dimensión N. Este servidor utilizará sockets para comunicarse con el cliente.

```
import java.io.*;
import java.net.*;

public class ServidorMatrices {

    public static void main(String[] args) {
        try {
            // Creamos el socket del servidor y lo ponemos en espera de conexiones
            ServerSocket servidor = new ServerSocket(9999);
            System.out.println("Servidor en espera de conexión...");

            // Aceptamos la conexión del cliente
            Socket cliente = servidor.accept();
            System.out.println("Cliente conectado");

            // Obtenemos los flujos de entrada y salida
            DataInputStream entrada = new DataInputStream(cliente.getInputStream());
            DataOutputStream salida = new DataOutputStream(cliente.getOutputStream());

            // Leemos la dimensión N
            int N = Integer.parseInt(entrada.readUTF());
            System.out.println("Dimensión N: " + N);

            // Creamos las matrices A y B
            int[][] A = new int[N][N];
            int[][] B = new int[N][N];

            // Leemos la matriz A
            for (int i = 0; i < N; i++) {
                for (int j = 0; j < N; j++) {
                    A[i][j] = Integer.parseInt(entrada.readUTF());
                }
            }

            // Leemos la matriz B
            for (int i = 0; i < N; i++) {
                for (int j = 0; j < N; j++) {
                    B[i][j] = Integer.parseInt(entrada.readUTF());
                }
            }

            // Calculamos la multiplicación A * B
            int[][] resultado = new int[N][N];
            for (int i = 0; i < N; i++) {
                for (int j = 0; j < N; j++) {
                    resultado[i][j] = 0;
                    for (int k = 0; k < N; k++) {
                        resultado[i][j] += A[i][k] * B[k][j];
                    }
                }
            }

            // Enviamos el resultado a través del socket
            for (int i = 0; i < N; i++) {
                for (int j = 0; j < N; j++) {
                    salida.writeUTF(resultado[i][j]);
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```
DataOutputStream salida = new DataOutputStream(cliente.getOutputStream());\n\n// Leemos la dimensión de las matrices\nint n = entrada.readInt();\nSystem.out.println("Dimensión de las matrices: " + n);\n\n// Creamos las matrices A y B\nint[][] a = new int[n][n];\nint[][] b = new int[n][n];\n\n// Leemos la matriz A\nSystem.out.println("Leyendo la matriz A...");\nfor (int i = 0; i < n; i++) {\n    for (int j = 0; j < n; j++) {\n        a[i][j] = entrada.readInt();\n    }\n}\n\n// Leemos la matriz B\nSystem.out.println("Leyendo la matriz B... ");\nfor (int i = 0; i < n; i++) {\n    for (int j = 0; j < n; j++) {\n        b[i][j] = entrada.readInt();\n    }\n}\n\n// Mostramos las matrices A y B en consola\nSystem.out.println("Matriz A:");\nmostrarMatriz(a);
```

```

System.out.println("Matriz B:");
mostrarMatriz(b);

// Cerramos los flujos y el socket del cliente
entrada.close();
salida.close();
cliente.close();

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

}

public static void mostrarMatriz(int[][] matriz) {
    for (int i = 0; i < matriz.length; i++) {
        for (int j = 0; j < matriz[0].length; j++) {
            System.out.print(matriz[i][j] + " ");
        }
        System.out.println();
    }
}

}

```

Ten en cuenta que este código simplemente recibe las matrices A y B del cliente y las muestra en consola. No realiza ningún cálculo con las matrices. Además, es necesario que el cliente envíe las matrices en el formato correcto y que ambos, cliente y servidor, estén de acuerdo en el protocolo de comunicación a seguir.

Mario Sebastian

modifica el programa para que la dimensión de la matrices sea N/3

ChatGPT

Aquí te proporciono la modificación del código anterior para que la dimensión de las matrices sea N/3:

```
import java.io.*;
import java.net.*;

public class ServidorMatrices {

    public static void main(String[] args) {
        try {
            // Creamos el socket del servidor y lo ponemos en espera de conexiones
            ServerSocket servidor = new ServerSocket(9999);
            System.out.println("Servidor en espera de conexión...");

            // Aceptamos la conexión del cliente
            Socket cliente = servidor.accept();
            System.out.println("Cliente conectado");

            // Obtenemos los flujos de entrada y salida
            DataInputStream entrada = new DataInputStream(cliente.getInputStream());
            DataOutputStream salida = new DataOutputStream(cliente.getOutputStream());

            // Leemos la dimensión de las matrices
            int n = entrada.readInt();
            System.out.println("Dimensión de las matrices: " + n);

            // Calculamos la dimensión de las submatrices
```

```
int m = n / 3;

// Creamos las matrices A y B
int[][] a = new int[m][m];
int[][] b = new int[m][m];

// Leemos la matriz A
System.out.println("Leyendo la matriz A...");
for (int i = 0; i < m; i++) {
    for (int j = 0; j < m; j++) {
        a[i][j] = entrada.readInt();
    }
}

// Leemos la matriz B
System.out.println("Leyendo la matriz B...");
for (int i = 0; i < m; i++) {
    for (int j = 0; j < m; j++) {
        b[i][j] = entrada.readInt();
    }
}

// Mostramos las matrices A y B en consola
System.out.println("Matriz A:");
mostrarMatriz(a);
System.out.println("Matriz B:");
mostrarMatriz(b);

// Cerramos los flujos y el socket del cliente
```

```

    entrada.close();
    salida.close();
    cliente.close();

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

}

public static void mostrarMatriz(int[][] matriz) {
    for (int i = 0; i < matriz.length; i++) {
        for (int j = 0; j < matriz[0].length; j++) {
            System.out.print(matriz[i][j] + " ");
        }
        System.out.println();
    }
}

}

```

Sin embargo, al correr el programa que escribió lanzaba un error, por lo tanto le mandé el error para que lo corrigiera

Mario Sebastian

tengo este error al correr el servidor: error: class, interface, enum, or record expected

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

^

ChatGPT

Este error puede ocurrir cuando una instrucción está fuera de lugar en el código, por lo que Java espera encontrar una clase, interfaz, enum o registro en ese punto en su lugar.

En el código que proporcioné anteriormente, puede ocurrir este error si la instrucción `int n = entrada.readInt();` se encuentra fuera de un método o bloque de código. Asegúrate de que esta instrucción esté dentro del método main:

```
public static void main(String[] args) {  
    try {  
        // Creamos el socket del servidor y lo ponemos en espera de conexiones  
        ServerSocket servidor = new ServerSocket(9999);  
        System.out.println("Servidor en espera de conexión...");  
  
        // Aceptamos la conexión del cliente  
        Socket cliente = servidor.accept();  
        System.out.println("Cliente conectado");  
  
        // Obtenemos los flujos de entrada y salida  
        DataInputStream entrada = new DataInputStream(cliente.getInputStream());  
        DataOutputStream salida = new DataOutputStream(cliente.getOutputStream());  
  
        // Leemos la dimensión de las matrices  
        int n = entrada.readInt(); // Esta instrucción debe estar dentro del método main  
        System.out.println("Dimensión de las matrices: " + n);  
  
        // Calculamos la dimensión de las submatrices  
        int m = n / 3;  
  
        // Creamos las matrices A y B  
        int[][] a = new int[m][m];  
        int[][] b = new int[m][m];  
  
        // Leemos la matriz A
```

```
System.out.println("Leyendo la matriz A...");

for (int i = 0; i < m; i++) {
    for (int j = 0; j < m; j++) {
        a[i][j] = entrada.readInt();
    }
}

// Leemos la matriz B

System.out.println("Leyendo la matriz B...");

for (int i = 0; i < m; i++) {
    for (int j = 0; j < m; j++) {
        b[i][j] = entrada.readInt();
    }
}

// Mostramos las matrices A y B en consola

System.out.println("Matriz A:");
mostrarMatriz(a);

System.out.println("Matriz B:");
mostrarMatriz(b);

// Cerramos los flujos y el socket del cliente

entrada.close();
salida.close();
cliente.close();

} catch (IOException e) {
    e.printStackTrace();
}
```

}

Si este no es el problema, revisa cuidadosamente el código y verifica que todas las llaves y paréntesis estén correctamente cerrados y abiertos en el lugar correcto.

Al final hacia falta una {

Luego al revisar los detalles de la práctica olvidé que las matrices no son de tipo int si no de tipo double, entonces le pedí que lo hiciera

Mario Sebastian

el siguiente código en java inicializa 2 matrices A, B, calcula la transpuesta de la matriz B y manda dichas matrices por un socket tcp, modifica el programa para que las matrices sean de tipo double en vez de int: **import java.io.*;**

```
import java.net.*;
```

```
public class Nodo0 {  
    public static void main(String[] args) {  
        int N = 4; // la dimensión de la matriz N*N  
        int[][] A = new int[N][N];  
        int[][] B = new int[N][N];  
  
        // Inicializar la matriz A  
        for(int i=0; i<N; i++) {  
            for(int j=0; j<N; j++) {  
                A[i][j] = 2*i+j;  
            }  
        }  
    }  
}
```

```

// Inicializar la matriz B y calcular su transpuesta

for(int i=0; i<N; i++) {
    for(int j=0; j<N; j++) {
        B[i][j] = 3*i-j;

        // Calcular la matriz transpuesta de B mientras se inicializa
        B[j][i] = 3*i-j;
    }
}

// Mostrar la matriz A en la consola
System.out.println("Matriz A:");
mostrarMatriz(A);

// Mostrar la matriz transpuesta de B en la consola
System.out.println("Matriz transpuesta de B:");
mostrarMatriz(B);

// Enviar las matrices al servidor
try {
    // Conectar al servidor
    Socket socket = new Socket("localhost", 5000);

    // Obtener los flujos de entrada y salida
    ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());
    ObjectInputStream in = new ObjectInputStream(socket.getInputStream());

    //Enviar dimension
    out.writeObject(N);
}

```

```
// Enviar la matriz A
out.writeObject(A);

// Enviar la matriz B
out.writeObject(B);

// Recibir la matriz transpuesta de B desde el servidor
int[][] B_transpuesta = (int[][])(in.readObject());

// Mostrar la matriz transpuesta de B recibida del servidor en la consola
System.out.println("Matriz transpuesta de B recibida del servidor:");
mostrarMatriz(B_transpuesta);

// Cerrar los flujos y el socket
out.close();
in.close();
socket.close();

} catch (IOException | ClassNotFoundException e) {
    e.printStackTrace();
}

}

// Método para mostrar una matriz en la consola
public static void mostrarMatriz(int[][] matriz) {
    for(int i=0; i<matriz.length; i++) {
        for(int j=0; j<matriz[0].length; j++) {
            System.out.print(matriz[i][j] + " ");
        }
    }
}
```

```
        System.out.println();  
    }  
  
    System.out.println();  
}  
}
```

ChatGPT

Para modificar el programa y hacer que las matrices sean de tipo double en lugar de int, debemos realizar los siguientes cambios:

- 1- Cambiar la definición de las matrices A y B de int[][] a double[][].
- 2- Actualizar los valores de las matrices A y B para que sean de tipo double en lugar de int.
- 3- Actualizar el cálculo de la matriz transpuesta de B para que se realice con valores de tipo double.
- 4- Actualizar el tipo de datos en el método mostrarMatriz de int[][] a double[][].
- 5- Actualizar el tipo de datos en la línea donde se recibe la matriz transpuesta de B desde el servidor de int[][] a double[][].

Aquí está el código actualizado con estos cambios:

```
import java.io.*;  
import java.net.*;  
  
public class Nodo0 {  
    public static void main(String[] args) {
```

```
int N = 4; // la dimensión de la matriz N*N
double[][] A = new double[N][N];
double[][] B = new double[N][N];

// Inicializar la matriz A
for(int i=0; i<N; i++) {
    for(int j=0; j<N; j++) {
        A[i][j] = 2*i+j;
    }
}

// Inicializar la matriz B y calcular su transpuesta
for(int i=0; i<N; i++) {
    for(int j=0; j<N; j++) {
        B[i][j] = 3*i-j;
    }
}

// Calcular la matriz transpuesta de B mientras se inicializa
B[j][i] = 3*i-j;
}

}

// Mostrar la matriz A en la consola
System.out.println("Matriz A:");
mostrarMatriz(A);

// Mostrar la matriz transpuesta de B en la consola
System.out.println("Matriz transpuesta de B:");
mostrarMatriz(B);
```

```
// Enviar las matrices al servidor

try {
    // Conectar al servidor
    Socket socket = new Socket("localhost", 5000);

    // Obtener los flujos de entrada y salida
    ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());
    ObjectInputStream in = new ObjectInputStream(socket.getInputStream());

    //Enviar dimension
    out.writeObject(N);

    // Enviar la matriz A
    out.writeObject(A);

    // Enviar la matriz B
    out.writeObject(B);

    // Recibir la matriz transpuesta de B desde el servidor
    double[][] B_transpuesta = (double[][])in.readObject();

    // Mostrar la matriz transpuesta de B recibida del servidor en la consola
    System.out.println("Matriz transpuesta de B recibida del servidor:");
    mostrarMatriz(B_transpuesta);

    // Cerrar los flujos y el socket
    out.close();
    in.close();
    socket.close();
}
```

```

} catch (IOException | ClassNotFoundException e) {
    e.printStackTrace();
}

}

// Método para mostrar una matriz en la consola
public static void mostrarMatriz(double[][] matriz) {
    for(int i=0; i<matriz.length; i++) {
        for(int j=0; j<matriz[0].length; j++) {
            System.out.print(matriz[i][j] + " ");
        }
        System.out.println();
    }
    System.out.println();
}
}

```

Hasta este punto ya tenemos la inicialización de matrices, el cliente que manda las matrices y el servidor que las recibe, sin embargo, al juntar las partes, en los sockets por donde se mandaban los datos, no se recibían correctamente o muchas veces se recibía basura. Esto se debía a que el cliente mandaba un tipo de dato y el servidor recibía otro tipo de dato. Así que le pregunte a ChatGPT para ver si lo podía resolver

Mario Sebastian

por que en el cliente mando un entero por medio de sockets TCP en java y el servidor recibe un número negativo?

ChatGPT

Hay varias posibles razones por las cuales esto podría estar sucediendo, pero sin ver el código específico es difícil determinar la causa exacta. Algunas de las posibles causas son:

Problemas de conversión de tipos: Puede ser que en el cliente se esté enviando un valor de tipo entero sin signo (es decir, un valor positivo), pero en el servidor se está recibiendo como un entero

con signo (que puede ser negativo). En este caso, es posible que deba realizarse una conversión explícita de tipos para asegurarse de que se transmita correctamente el valor.

Problemas con la representación binaria: Los datos se envían a través de la red en formato binario, y es posible que haya un problema con la forma en que se está codificando o decodificando el entero en cada extremo. Por ejemplo, el cliente podría estar usando un byte-ordering diferente al del servidor, lo que resultaría en un número negativo cuando se recibe en el servidor.

Problemas en el protocolo: Dependiendo del protocolo que se esté utilizando, puede haber ciertos campos reservados para indicar el tipo de datos que se están enviando. Si estos campos no se están configurando correctamente en el cliente, es posible que el servidor interprete los datos de manera incorrecta.

Para solucionar este problema, se recomienda revisar cuidadosamente el código del cliente y el servidor para verificar si hay errores de conversión o problemas de protocolo. También puede ser útil utilizar herramientas de depuración y monitoreo para identificar el problema con mayor precisión

Mario Sebastian

este es el código del cliente: **import java.io.*;**

import java.net.*;

```
public class Nodo0 {  
    public static void main(String[] args) {  
        int N = 4; // la dimensión de la matriz N*N  
        double[][] A = new double[N][N];  
        double[][] B = new double[N][N];  
  
        // Inicializar la matriz A  
        for(int i=0; i<N; i++) {  
            for(int j=0; j<N; j++) {  
                A[i][j] = 2*i+j;  
            }  
        }  
    }  
}
```

```
    }

}

// Inicializar la matriz B y calcular su transpuesta

for(int i=0; i<N; i++) {

    for(int j=0; j<N; j++) {

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

        // Calcular la matriz transpuesta de B mientras se inicializa

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

    }

}

// Mostrar la matriz A en la consola

System.out.println("Matriz A:");

mostrarMatriz(A);

// Mostrar la matriz transpuesta de B en la consola

System.out.println("Matriz transpuesta de B:");

mostrarMatriz(B);

// Enviar las matrices al servidor

try {

    // Conectar al servidor

    Socket socket = new Socket("localhost", 5000);

    // Obtener los flujos de entrada y salida

    ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());

    ObjectInputStream in = new ObjectInputStream(socket.getInputStream());
```

```
//Enviar dimension
out.writeInt(N);

// Enviar la matriz A
out.writeObject(A);

// Enviar la matriz B
out.writeObject(B);

// Recibir la matriz transpuesta de B desde el servidor
double[][] B_transpuesta = (double[][])(in.readObject());

// Mostrar la matriz transpuesta de B recibida del servidor en la consola
System.out.println("Matriz transpuesta de B recibida del servidor:");
mostrarMatriz(B_transpuesta);

// Cerrar los flujos y el socket
out.close();
in.close();
socket.close();

} catch (IOException | ClassNotFoundException e) {
    e.printStackTrace();
}

}

// Método para mostrar una matriz en la consola
public static void mostrarMatriz(double[][] matriz) {
    for(int i=0; i<matriz.length; i++) {
        for(int j=0; j<matriz[0].length; j++) {
```

```
        System.out.print(matriz[i][j] + " ");
    }

    System.out.println();
}

System.out.println();
}

}

este es el código del servidor: import java.io.*;
import java.net.*;

public class Servidor{

public static void main(String[] args) {
    try {
        // Creamos el socket del servidor y lo ponemos en espera de conexiones
        ServerSocket servidor = new ServerSocket(5000);
        System.out.println("Servidor en espera de conexión...");

        // Aceptamos la conexión del cliente
        Socket cliente = servidor.accept();
        System.out.println("Cliente conectado");

        // Obtenemos los flujos de entrada y salida
        DataInputStream entrada = new DataInputStream(cliente.getInputStream());
        DataOutputStream salida = new DataOutputStream(cliente.getOutputStream());

        // Leemos la dimensión de las matrices
        int n = entrada.readInt();
        System.out.println("Dimensión de las matrices: " + n);
    }
}
```

```
// Calculamos la dimensión de las submatrices  
int m = n / 3;  
  
// Creamos las matrices A y B  
double[][] a = new double[m][m];  
double[][] b = new double[m][m];  
  
// Leemos la matriz A  
System.out.println("Leyendo la matriz A...");  
for (int i = 0; i < m; i++) {  
    for (int j = 0; j < m; j++) {  
        a[i][j] = entrada.readInt();  
    }  
}  
  
// Leemos la matriz B  
System.out.println("Leyendo la matriz B...");  
for (int i = 0; i < m; i++) {  
    for (int j = 0; j < m; j++) {  
        b[i][j] = entrada.readInt();  
    }  
}  
  
// Mostramos las matrices A y B en consola  
System.out.println("Matriz A:");  
mostrarMatriz(a);  
System.out.println("Matriz B:");  
mostrarMatriz(b);
```

```
// Cerramos los flujos y el socket del cliente
entrada.close();
salida.close();
cliente.close();

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

}

public static void mostrarMatriz(double[][] matriz) {
    for (int i = 0; i < matriz.length; i++) {
        for (int j = 0; j < matriz[0].length; j++) {
            System.out.print(matriz[i][j] + " ");
        }
        System.out.println();
    }
}
```

ChatGPT

El problema en este caso es que en el servidor se está leyendo los datos de las matrices como enteros utilizando el método `readInt()`, pero en el cliente se están enviando las matrices como objetos utilizando el método `writeObject()`. Esto causa una discrepancia en el tipo de datos esperado y leído en el servidor, lo que puede llevar a que los valores se interpreten de forma incorrecta.

Para solucionar este problema, se debe utilizar el método `readObject()` en el servidor para leer las matrices como objetos, y luego realizar la conversión correspondiente a un arreglo de dobles. En el cliente, se deben enviar las matrices como objetos utilizando el método `writeObject()`.

Mario Sebastian

cómo se vería en código?

ChatGPT

En el código del servidor, el cliente está enviando un objeto de tipo `double[][]` para la matriz A y la matriz B a través de un `ObjectOutputStream`. Sin embargo, en el servidor, se están leyendo las matrices como una secuencia de enteros a través de un `DataInputStream`, lo cual no es compatible y es la causa del problema que estás experimentando.

Para solucionar este problema, el servidor también debe leer las matrices como objetos de tipo `double[][]`. Aquí te dejo una posible solución:

```
import java.io.*;
import java.net.*;

public class Servidor{

    public static void main(String[] args) {
        try {
            // Creamos el socket del servidor y lo ponemos en espera de conexiones
            ServerSocket servidor = new ServerSocket(5000);
            System.out.println("Servidor en espera de conexión...");

            // Aceptamos la conexión del cliente
            Socket cliente = servidor.accept();
            System.out.println("Cliente conectado");

            // Obtenemos los flujos de entrada y salida
            ObjectInputStream entrada = new ObjectInputStream(cliente.getInputStream());
            double[][] matrizA = (double[][]) entrada.readObject();
            double[][] matrizB = (double[][]) entrada.readObject();

            // Realizamos la multiplicación de matrices
            double[][] resultado = multiplicarMatrices(matrizA, matrizB);

            ObjectOutputStream salida = new ObjectOutputStream(cliente.getOutputStream());
            salida.writeObject(resultado);
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

    private static double[][] multiplicarMatrices(double[][] A, double[][] B) {
        int n = A.length;
        int m = A[0].length;
        int p = B[0].length;

        double[][] resultado = new double[n][p];
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < p; j++) {
                for (int k = 0; k < m; k++) {
                    resultado[i][j] += A[i][k] * B[k][j];
                }
            }
        }
        return resultado;
    }
}
```

```
ObjectOutputStream salida = new ObjectOutputStream(cliente.getOutputStream());
ObjectInputStream entrada = new ObjectInputStream(cliente.getInputStream());

// Leemos la dimensión de las matrices
int n = entrada.readInt();
System.out.println("Dimensión de las matrices: " + n);

// Calculamos la dimensión de las submatrices
int m = n / 3;

// Creamos las matrices A y B
double[][] a = (double[][])entrada.readObject();
double[][] b = (double[][])entrada.readObject();

// Mostramos las matrices A y B en consola
System.out.println("Matriz A:");
mostrarMatriz(a);
System.out.println("Matriz B:");
mostrarMatriz(b);

// Cerramos los flujos y el socket del cliente
entrada.close();
salida.close();
cliente.close();

} catch (IOException | ClassNotFoundException e) {
    e.printStackTrace();
}

}
```

```

public static void mostrarMatriz(double[][] matriz) {
    for (int i = 0; i < matriz.length; i++) {
        for (int j = 0; j < matriz[0].length; j++) {
            System.out.print(matriz[i][j] + " ");
        }
        System.out.println();
    }
}

```

En el código del servidor, se cambió la creación de flujos de entrada y salida a ObjectOutputStream y ObjectInputStream, respectivamente. También se cambió la forma de leer las matrices A y B, ahora se están leyendo como objetos de tipo double[][] a través del método readObject() del ObjectInputStream.

Teniendo es parte solucionada, pasamos a la parte de la división de matrices en 3 partes. En otra conversación le pedí lo siguiente:

Mario Sebastian

el siguiente programa en java inicializa 2 matrices A, B, obtiene la transpuesta de B y manda dichas matrices por un socket TCP, modifica el código para obtener 3 submatrices a partir de la matriz A y B transpuesta, es decir vas a dividir la matriz A y B en 3 filas, cada división será una nueva submatriz A1,A2,A3. Lo mismo para B, cada 3 filas será una nueva submatriz Bt1,Bt2,Bt3. El código es el siguiente: **import java.io.*;**

```
import java.net.*;
```

```

public class Nodo0 {
    public static void main(String[] args) {
        int N = 4; // la dimensión de la matriz N*N
        double[][] A = new double[N][N];
        double[][] B = new double[N][N];
    }
}
```

```
// Inicializar la matriz A
for(int i=0; i<N; i++) {
    for(int j=0; j<N; j++) {
        A[i][j] = 2*i+j;
    }
}

// Inicializar la matriz B y calcular su transpuesta
for(int i=0; i<N; i++) {
    for(int j=0; j<N; j++) {
        B[i][j] = 3*i-j;
    }
}

// Calcular la matriz transpuesta de B mientras se inicializa
B[j][i] = 3*i-j;
}

}

// Mostrar la matriz A en la consola
System.out.println("Matriz A:");
mostrarMatriz(A);

// Mostrar la matriz transpuesta de B en la consola
System.out.println("Matriz transpuesta de B:");
mostrarMatriz(B);

// Enviar las matrices al servidor
try {
    // Conectar al servidor
    Socket socket = new Socket("localhost", 5000);
```

```
// Obtener los flujos de entrada y salida
ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());
ObjectInputStream in = new ObjectInputStream(socket.getInputStream());

//Enviar dimension
out.writeInt(N);

// Enviar la matriz A
out.writeObject(A);

// Enviar la matriz B
out.writeObject(B);

// Recibir la matriz transpuesta de B desde el servidor
double[][] B_transpuesta = (double[][])(in.readObject());

// Mostrar la matriz transpuesta de B recibida del servidor en la consola
System.out.println("Matriz transpuesta de B recibida del servidor:");
mostrarMatriz(B_transpuesta);

// Cerrar los flujos y el socket
out.close();
in.close();
socket.close();

} catch (IOException | ClassNotFoundException e) {
    e.printStackTrace();
}

}
```

```

// Método para mostrar una matriz en la consola
public static void mostrarMatriz(double[][] matriz) {
    for(int i=0; i<matriz.length; i++) {
        for(int j=0; j<matriz[0].length; j++) {
            System.out.print(matriz[i][j] + " ");
        }
        System.out.println();
    }
    System.out.println();
}

```

Para dividir la matriz A y B transpuesta en tres submatrices cada una, podemos modificar el código de la siguiente manera:

```

import java.io.*;
import java.net.*;

public class Nodo0 {
    public static void main(String[] args) {
        int N = 4; // la dimensión de la matriz N*N
        double[][] A = new double[N][N];
        double[][] B = new double[N][N];

        // Inicializar la matriz A
        for(int i=0; i<N; i++) {
            for(int j=0; j<N; j++) {
                A[i][j] = 2*i+j;
            }
        }
    }
}

```

```

// Inicializar la matriz B y calcular su transpuesta

for(int i=0; i<N; i++) {
    for(int j=0; j<N; j++) {
        B[i][j] = 3*i-j;
    }
}

// Calcular la matriz transpuesta de B mientras se inicializa
B[j][i] = 3*i-j;
}

}

// Mostrar la matriz A en la consola
System.out.println("Matriz A:");
mostrarMatriz(A);

// Mostrar la matriz transpuesta de B en la consola
System.out.println("Matriz transpuesta de B:");
mostrarMatriz(B);

// Dividir la matriz A en 3 submatrices
double[][] A1 = new double[N/3][N];
double[][] A2 = new double[N/3][N];
double[][] A3 = new double[N/3][N];
for(int i=0; i<N; i++) {
    if(i < N/3) {
        A1[i] = A[i];
    } else if(i < 2*N/3) {
        A2[i-N/3] = A[i];
    } else {
        A3[i-2*N/3] = A[i];
    }
}

```

```

    }

}

// Dividir la matriz transpuesta de B en 3 submatrices

double[][] Bt = new double[N][N];

for(int i=0; i<N; i++) {

    for(int j=0; j<N; j++) {

        Bt[i][j] = B[j][i];
    }
}

double[][] Bt1 = new double[N/3][N];

double[][] Bt2 = new double[N/3][N];

double[][] Bt3 = new double[N/3][N];

for(int i=0; i<N; i++) {

    if(i < N/3) {

        Bt1[i] = Bt[i];
    } else if(i < 2*N/3) {

        Bt2[i-N/3] = Bt[i];
    } else {

        Bt3[i-2*N/3] = Bt[i];
    }
}

// Enviar las matrices al servidor

try {

    // Conectar al servidor

    Socket socket = new Socket("localhost", 5000);

    // Obtener los flujos de entrada y salida
}

```

```
ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());  
ObjectInputStream in = new ObjectInputStream(socket.getInputStream());  
  
//Enviar dimension  
out.writeInt(N);  
  
// Enviar la submatriz A1  
out.writeObject(A1);  
  
// Enviar la submatriz A2  
out.writeObject(A2);  
  
// Enviar la submatriz A3  
out.writeObject(A3);
```

En esta parte se traba porque el código fue algo largo y no le alcanzaron los tokens, sin embargo, la parte que nos interesaba esta al principio y si la alcanzó a escribir, por lo tanto, solamente copie los fragmentos clave del código y lo agregue al código base que ya se tenía.

Ahora nos faltaba la parte de enviar las matrices que tenemos divididas a 3 servidores, en otra conversación le pedí lo siguiente:

Mario Sebastian

tengo un cliente en java que usa sockets TCP, quiero enviar datos a 3 servidores al mismo tiempo, que debo hacer?

ChatGPT

Para enviar datos a tres servidores al mismo tiempo en Java utilizando sockets TCP, puedes crear tres instancias de la clase Socket y conectarlas a los servidores correspondientes. Luego, puedes enviar los datos a cada uno de los servidores utilizando los métodos getOutputStream() y write() de cada una de las instancias de Socket.

Aquí te dejo un ejemplo básico que ilustra cómo podrías enviar datos a tres servidores utilizando threads para hacerlo simultáneamente:

```
import java.io.*;
import java.net.*;

public class ClienteTCP {
    public static void main(String[] args) {
        String servidor1 = "localhost";
        int puerto1 = 8080;
        String servidor2 = "localhost";
        int puerto2 = 8081;
        String servidor3 = "localhost";
        int puerto3 = 8082;

        try {
            // Conectarse a los servidores
            Socket socket1 = new Socket(servidor1, puerto1);
            Socket socket2 = new Socket(servidor2, puerto2);
            Socket socket3 = new Socket(servidor3, puerto3);

            // Crear threads para enviar datos simultáneamente
            Thread t1 = new Thread(new Runnable() {
                public void run() {
                    try {
                        OutputStream outputStream = socket1.getOutputStream();
                        outputStream.write("Datos para el servidor 1".getBytes());
                        outputStream.flush();
                        socket1.close();
                    } catch (IOException e) {
                }
            });
        }
    }
}
```

```
        e.printStackTrace();
    }
}

});

Thread t2 = new Thread(new Runnable() {
    public void run() {
        try {
            OutputStream outputStream = socket2.getOutputStream();
            outputStream.write("Datos para el servidor 2".getBytes());
            outputStream.flush();
            socket2.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
};

Thread t3 = new Thread(new Runnable() {
    public void run() {
        try {
            OutputStream outputStream = socket3.getOutputStream();
            outputStream.write("Datos para el servidor 3".getBytes());
            outputStream.flush();
            socket3.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
});
```

```
});

// Iniciar los threads
t1.start();
t2.start();
t3.start();

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

}

}

}
```

Este ejemplo crea tres instancias de Socket para conectarse a los servidores en los puertos 8080, 8081 y 8082 respectivamente. Luego, se crean tres threads que envían datos a cada uno de los servidores utilizando sus respectivas instancias de Socket. Los threads se inician utilizando el método start() y corren simultáneamente.

Es importante destacar que en este ejemplo se están cerrando las conexiones a los servidores después de enviar los datos. Si necesitas mantener la conexión abierta para recibir una respuesta del servidor, deberás ajustar el código en consecuencia.

Posteriormente pasamos al algoritmo de la multiplicación, para esto se lo pedí de la siguiente manera tratando de no utilizar la palabra matriz debido a que no tuviera la idea de que estuviéramos trabajando con matrices, en su lugar utilice la palabra arreglo bidimensional.

Mario Sebastian

por favor haz un programa en java que multiplique los elementos de 2 arreglos bidimensionales de tamaño $N/3 \times N$, sea $A[i][j]$ y $B[i][j]$ elementos de los arreglos, la multiplicación será $A[i][j] * B[i][j]$

ChatGPT

Por supuesto, aquí tienes un programa en Java que multiplica los elementos de dos arreglos bidimensionales de tamaño $N/3 * N$ y almacena el resultado en un tercer arreglo:

```
import java.util.Random;

public class MultiplicacionArreglos {

    public static void main(String[] args) {

        // Definir el tamaño del arreglo
        int n = 9; // Debe ser divisible por 3

        // Inicializar los arreglos A y B con valores aleatorios
        double[][] A = new double[n/3][n];
        double[][] B = new double[n/3][n];
        Random rand = new Random();

        for (int i = 0; i < n/3; i++) {
            for (int j = 0; j < n; j++) {
                A[i][j] = rand.nextDouble() * 10.0;
                B[i][j] = rand.nextDouble() * 10.0;
            }
        }

        // Imprimir los arreglos A y B
        System.out.println("Arreglo A:");
        imprimirArreglo(A);
        System.out.println("Arreglo B:");
        imprimirArreglo(B);

        // Multiplicar los elementos de A y B y almacenar el resultado en C
    }
}
```

```

double[][] C = new double[n/3][n];
for (int i = 0; i < n/3; i++) {
    for (int j = 0; j < n; j++) {
        C[i][j] = A[i][j] * B[i][j];
    }
}

// Imprimir el arreglo resultante C
System.out.println("Arreglo C:");
imprimirArreglo(C);
}

// Método para imprimir un arreglo bidimensional
public static void imprimirArreglo(double[][] arr) {
    for (int i = 0; i < arr.length; i++) {
        for (int j = 0; j < arr[i].length; j++) {
            System.out.print(arr[i][j] + " ");
        }
        System.out.println();
    }
    System.out.println();
}
}

```

Sin embargo, estaba mal porque infirió que estábamos multiplicando matrices y en vez de hacerlo renglón por renglón lo hacía columna por columna, esa parte la modifique para que fuera renglón por renglón.

Al final hice algunas modificaciones extras al código que ya se tenía.

Conclusiones

En esta práctica aprendí sobre la localidad espacial y sobre cómo se almacenan los datos en memoria, dado que se almacenan de forma secuencial. es más rápido multiplicar los renglones de las matrices a multiplicar renglón por columna computacionalmente hablando. También se me hizo muy interesante y padre utilizar Azure para correr el programa en 3 máquinas virtuales debido a que es más realista y aprendemos cosas útiles para el mundo laboral en vez de ejecutarlo todo en localhost.

También me di cuenta que chatGPT es muy útil si lo sabemos usar, es decir no esperar a que nos resuelva un problema grande con un simple prompt si no modularlo y entender bien el problema para explicarlo y obtener una solución lo más acertada posible. Considero que esto es el principio de la programación en lenguaje natural.