



# Instituto Politécnico Nacional

Escuela Superior de Cómputo

Sistemas Distribuidos

Pineda Guerrero Carlos

Tarea 5

4CV13

Humberto Alejandro Ortega Alcocer (2016630495)

2 de Mayo del 2023

# Índice

<b>Índice</b>	<b>1</b>
<b>Objetivo</b>	<b>2</b>
<b>Desarrollo</b>	<b>4</b>
Creación de Máquinas Virtuales	4
Verificando que tengamos acceso a la máquina virtual.	8
Configurando el servidor para la tarea.	9
Instalando Java 11	9
Copiando los archivos necesarios.	11
Creación del resto de la topología	12
Creación de Código (ChatGPT)	13
Ejecución del Programa	23
Primer caso	24
Segundo caso	24
<b>Conclusión</b>	<b>25</b>

## Objetivo

Utilizando ChatGPT se deberá desarrollar un sistema que calcule el producto de dos matrices rectangulares utilizando Java RMI, tal como se explicó en clase.

Sea la matriz A con dimensiones NxM, y la matriz B con dimensiones MxN.

Sea C = Ax B.

Se deberán ejecutar dos casos:

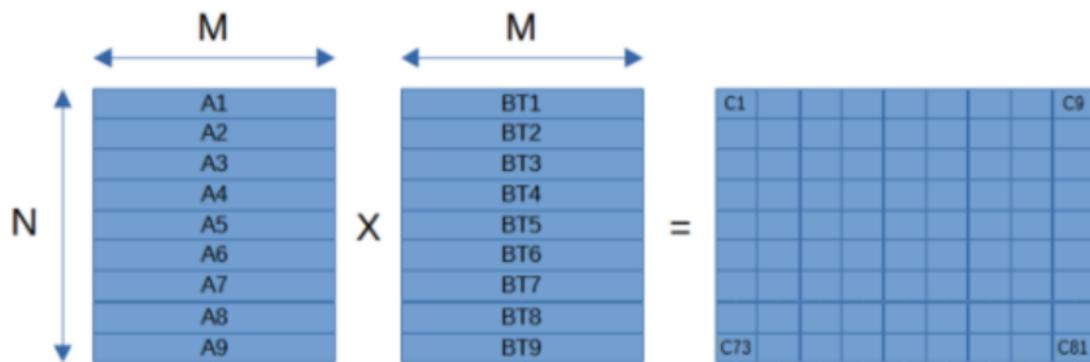
- N=9, M=4 se deberá desplegar la matriz C y el checksum de la matriz C.
- N=900, M=400 se deberá desplegar el checksum de la matriz C.
- Los elementos de las matrices A, B y C deberán ser de tipo float y el checksum deberá ser de tipo double.

Se deberá inicializar las matrices A y B de la siguiente manera:

`A[i][j]= 2 \* i + 3 \* j`

`B[i][j] = 3 \* i - 2 \* j`

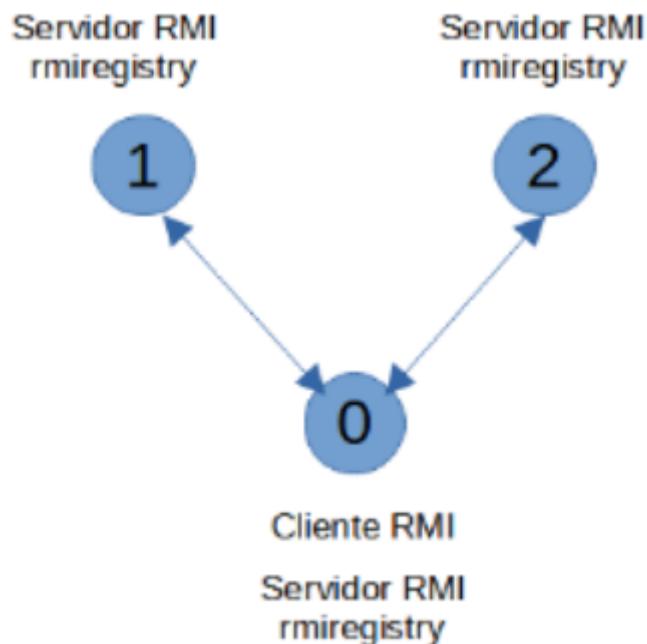
Sea BT la transpuesta de la matriz B.



Se deberá dividir las matrices A y BT en nueve partes iguales, por tanto la matriz C estará dividida en 81 partes:

La multiplicación de las matrices  $A_i \times B_{j'}$  deberá realizarse renglón por renglón, tal como vimos en la clase "Jerarquía de memoria".

Se deberá crear tres máquinas virtuales con Ubuntu en el mismo grupo de recursos, de acuerdo a la siguiente topología:



La máquina virtual 0 (nodo 0) ejecutará el cliente RMI el cual hará los siguiente:

- Inicializar las matrices A y B.
- Obtener la matriz BT.
- Obtener las matrices  $A_i$  ( $i=1,2,\dots,9$ ) y  $B_{Ti}$  ( $i=1,2,\dots,9$ )
- Utilizando RMI obtener las matrices  $C_1, C_2, \dots, C_{27}$  invocando el método `multiplica_matrices()` el cual ejecutará en el nodo 0.
- Utilizando RMI obtener las matrices  $C_{28}, C_{29}, \dots, C_{54}$  invocando el método `multiplica_matrices()` el cual ejecutará en el nodo 1.
- Utilizando RMI obtener las matrices  $C_{55}, C_{56}, \dots, C_{81}$  invocando el método `multiplica_matrices()` el cual ejecutará en el nodo 2.
- Obtener la matriz C a partir de las matrices  $C_1, C_2, \dots, C_{81}$ .
- Para el caso  $N=9$  y  $M=4$  desplegar la matriz C.
- Calcular y desplegar el checksum de la matriz C.
- Los nodos 0, 1 y 2 ejecutarán el servidor RMI y el rmiregistry.

El cliente RMI deberá utilizar threads para invocar el método remoto `multiplica_matrices()` de manera que los servidores RMI calculen los productos en paralelo.

# Desarrollo

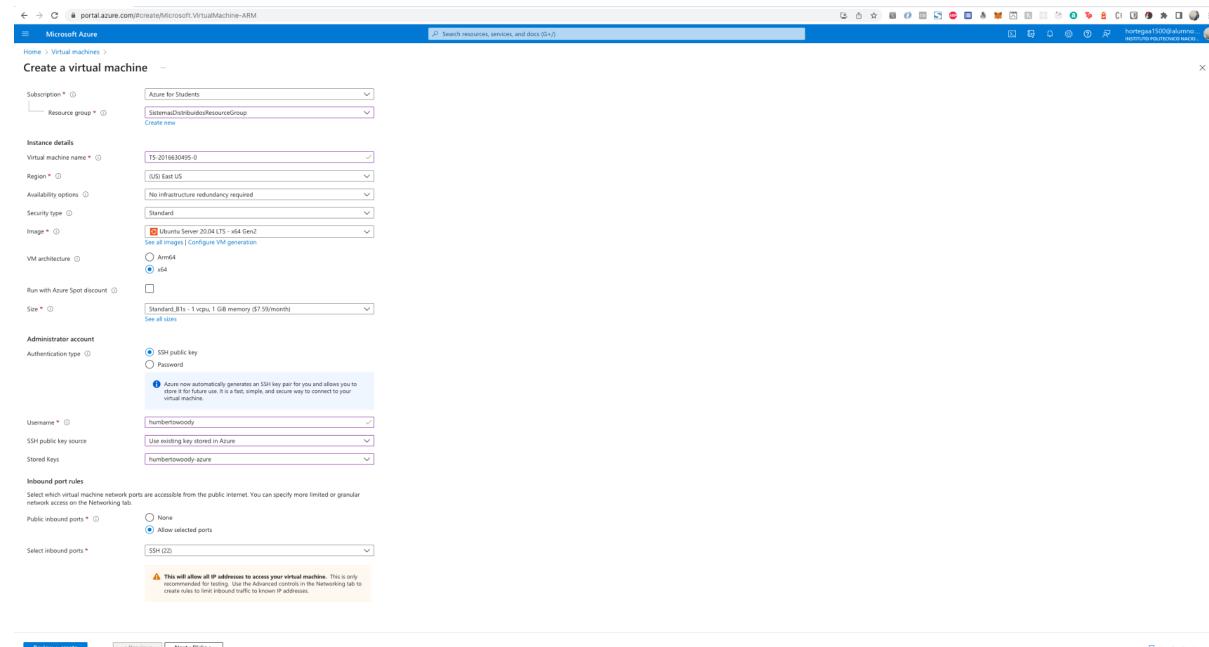
Para desarrollar esta tarea, se descompondrá en tres etapas: creación de máquina virtual, creación de código (usando ChatGPT) y pruebas de ejecución. Esto con el fin de profundizar en cada sección de forma adecuada.

## Creación de Máquinas Virtuales

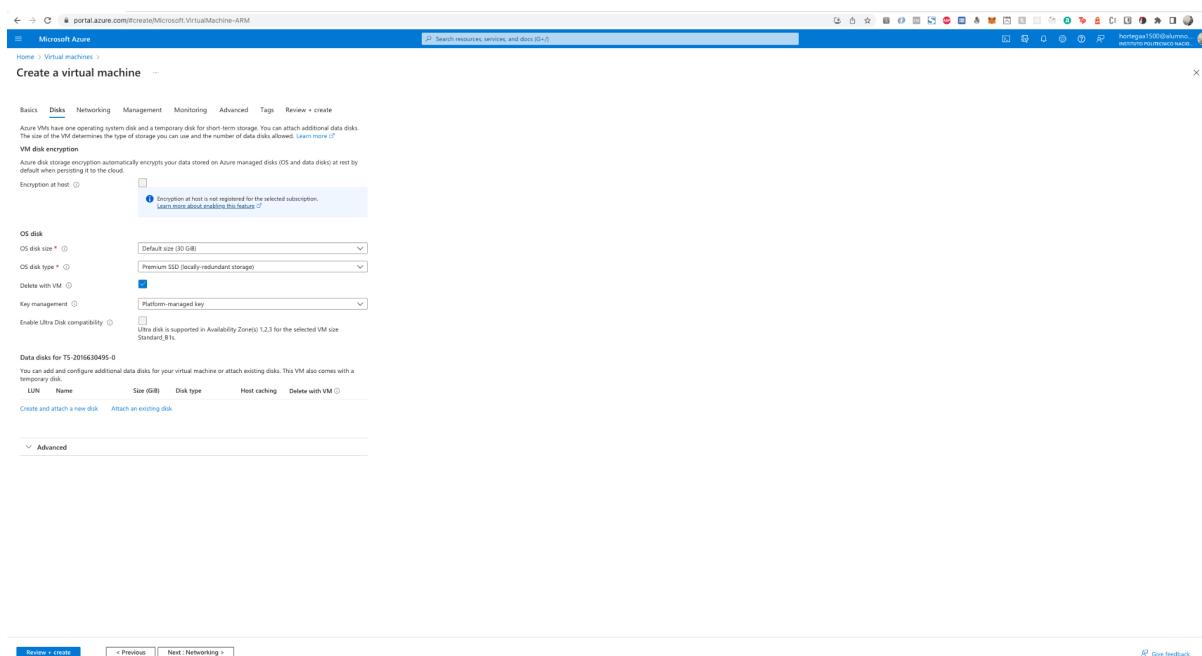
En esta sección primero visualizaremos paso a paso como se crea el primer nodo de nuestra topología, y posteriormente se mostrarán los otros dos nodos con la misma configuración. Lo primero será seguir el asistente para crear una máquina virtual nueva:



Lo siguiente será colocar la configuración de tamaño (1 vCPU y 1GB en RAM), así como las opciones de seguridad: la clave creada para la Tarea 1 (reutilizamos el mismo combo) y permitimos el tráfico TCP al puerto 22 para las sesiones de SSH (SCP, SFTP, etc).

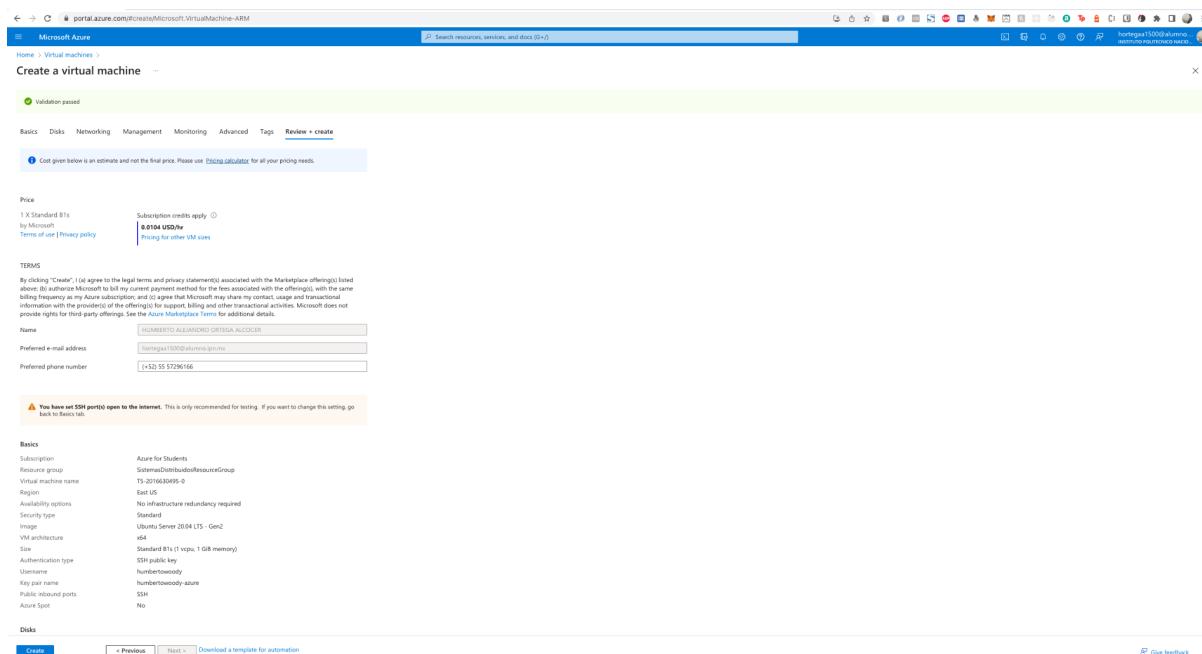


Lo siguiente será colocar la configuración de disco: HDD con 30GiB y sin redundancia.



The screenshot shows the 'Create a virtual machine' wizard in the Microsoft Azure portal. The 'Basic' step is active. Configuration for the OS disk is set to Standard B1s (30 GiB) and Premium SSD (locally-redundant storage). A note states that Ultra disk compatibility is supported. The 'Data disks' section is collapsed. At the bottom, there are 'Review + create' and 'Next: Networking >' buttons.

Finalmente, el resumen de nuestra máquina virtual.



The screenshot shows the 'Create a virtual machine' wizard in the 'Review + create' step. It summarizes the configuration: 1 X Standard B1s, 30 GiB, Premium SSD. A note states that SSH port is open. The 'TERMS' section is expanded, showing legal agreements. The 'Disk' section is collapsed. At the bottom, there are 'Review + create' and 'Create' buttons.

The screenshot shows the 'Create a virtual machine' wizard in the Microsoft Azure portal. The 'Networking' step is currently active. It includes fields for:

- Private network:** Internal (10.0.0.0/16)
- Subnet:** (new) (10.0.0.0/24)
- Public IP:** (new) TS-2016630495-0-ip
- Accelerated networking:** Off
- Place this virtual machine behind an application load balancing instance:** No
- Delete public IP and NIC when VM is deleted:** Disabled

Other sections shown in the wizard include:

- Validation passed**: Shows 'Success' status.
- Disk**: OS disk size (20 GB), Premium SSD (LRS).
- Management**: Monitor Defender for Cloud (None), System assigned managed identity (Off), Login with Azure AD (Off), Auto shutdown (Off), Enable hotpatch (Off), Patch orchestration options (Image Default).
- Monitoring**: Alerts (Off), Boot diagnostics (On), Enable OS guest diagnostics (Off).
- Advanced**: Driver (None), VM applications (None), Circular log (No), User data (None), Disk controller type (SCSI), Proximity placement group (None), Capacity reservation group (None).

At the bottom, there are 'Create' and 'Next >' buttons, along with a 'Download a template for automation' link and a 'Give feedback' button.

Creamos la máquina virtual.

The screenshot shows the 'CreateVm-canonical.0001-com-ubuntu-server-focal-2-20230501125901 | Overview' page in the Microsoft Azure portal. Key information includes:

- Deployment name:** CreateVm-canonical.0001-com-ubuntu-server-focal-2-20230501125901
- Subscription:** Azure for Students
- Resource group:** SistemaDistribuidosResourceGroup
- Start time:** 5/1/2023, 1:03:13 PM
- Correlation ID:** 15669c3d-a7d5-4a2b-ac16-21a4bf19957e

**Deployment details:**

Resource	Type	Status	Operation details
TS-2016630495-0	Microsoft.Compute/virtualMachines	OK	<a href="#">Operation details</a>
TS-2016630495-0-s12	Microsoft.Network/networkInterfaces	Created	<a href="#">Operation details</a>
TS-2016630495-0-vnet	Microsoft.Network/virtualNetworks	OK	<a href="#">Operation details</a>
TS-2016630495-0-ip	Microsoft.Network/publicIPAddresses	OK	<a href="#">Operation details</a>
TS-2016630495-0-nsg	Microsoft.Network/networkSecurityGroups	OK	<a href="#">Operation details</a>

**Next steps:**

- Setup auto shutdown - Recommended
- Monitor VM health, performance and network dependencies - Recommended
- Run a script inside the virtual machine - Recommended

Buttons at the bottom include 'Go to resource', 'Create another VM', 'Give feedback', and 'Tell us about your experience with deployment'.

**Right sidebar:**

- Cost Management:** Get notified to stay within your budget and prevent unexpected charges on your bill. [Set up cost alerts >](#)
- Microsoft Defender for Cloud:** Secure your apps and infrastructure. [Go to Microsoft Defender for Cloud >](#)
- Free Microsoft tutorials:** Start learning today! [>](#)
- Work with an expert:** Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support. [Find an Azure expert >](#)

Verificamos configuración final de la máquina virtual (para ver la IP pública y detalles finales).

The screenshot displays the Microsoft Azure portal interface for managing a virtual machine. The URL in the address bar is <https://portal.azure.com/#@correo.ipn.mx/resource/subscriptions/a5d9d875-3ae1-4e2b-95b5-02150d67b8d2/resourceGroups/SistemasDistribuidos/providers/Microsoft.Compute/virtualMachines/TS-2016630495-0/overview>. The main content area shows the following details for the virtual machine 'TS-2016630495-0':

- Essentials:**
  - Resource group: SistemasDistribuidosResourceGroup
  - Status: Running
  - Location: East US
  - Subscription: Aguirre for Students
  - Subscription ID: a5d9d875-3ae1-4e2b-95b5-02150d67b8d2
  - Tags: None
  - DNS name: TS-2016630495-0.vnet.default
  - Health state: -
- Properties:**
  - Computer name: TS-2016630495-0
  - Operating system: Ubuntu (Ubuntu 20.04)
  - Publisher: canonical
  - Offer: 0001-com-ubuntu-server-focal
  - Plan: 20-04-its-gen2
  - VM generation: V2
  - VM architecture: x64
  - Agent status: Ready
  - Agent version: 2.9.4
  - Host group: None
  - Host: -
  - Priority placement group: -
  - Collection status: N/A
  - Capacity reservation group: -
- Availability + scaling:**
  - Availability zone: -
  - Availability set: -
  - Scale Set: -
- Security type:** Standard
- Extensions + applications:** None
- Networking:**
  - Public IP address: 172.190.15.187 (Network interface 0-2016630495-0vif12)
  - Private IP address: 10.0.0.4
  - Private IP address (IPv6): -
  - Virtual network/subnet: TS-2016630495-0.vnet/default
  - DNS name: Configure
- Disk:**
  - OS disk: TS-2016630495-0\_OsDisk\_1\_9164c237dd941289a404af1d1d8d558
  - Encryption at host: Disabled
  - Encrypted disk encryption: Not enabled
  - Ephemeral OS disk: N/A
  - Data disks: 0
- Auto-shutdown:** Not enabled
- Azure Spot:** None

## Verificando que tengamos acceso a la máquina virtual.

Para verificar que tenemos acceso a la máquina virtual, nos conectaremos utilizando las llaves previamente configuradas en la Tarea #1.

```
> ls
humbertowoody-azure.pem
> ssh -i ./humbertowoody-azure.pem humbertowoody@172.190.15.187
The authenticity of host '172.190.15.187 (172.190.15.187)' can't be established.
ED25519 key fingerprint is SHA256:odmoChf40Q5cQx97UPPIBBofimtf0Gx/HaoHSrGz6ck.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.190.15.187' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1037-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon May  1 19:05:53 UTC 2023

System load:  0.17          Processes:      106
Usage of /:   5.2% of 28.89GB  Users logged in:  0
Memory usage: 29%           IPv4 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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

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.

humbertowoody@T5-2016630495-0:~$ echo "funcionaaaaa wuuuuu!"
funcionaaaaa wuuuuu!
humbertowoody@T5-2016630495-0:~$ █
```

## Configurando el servidor para la tarea.

Para poder comenzar a ejecutar nuestra tarea, necesitaremos realizar tres acciones en cada servidor (máquina virtual Ubuntu 20.04) que nos permitan ejecutar nuestro programa así como comunicarnos con el mismo remotamente.

Lo primero será instalar Java dado que nuestra tarea se ha realizado en dicho lenguaje de programación y Ubuntu (20.04) no incluye una versión del JDK ni JRE en el sistema operativo por defecto.

Finalmente, deberemos copiar el código fuente del Servidor, Cliente, Interfaz e Implementación a cada máquina para ejecutar los casos solicitados por el profesor.

Instalando Java 11

Se ha determinado utilizar Java en su versión 11 porque es lo que tengo instalado en mi computadora local y no quisiera que hubiera incompatibilidades entre las versiones, esto a pesar de que las funcionalidades que utilizamos (sockets seguros) no han cambiado desde Java 8.

Para instalar Java 11 en Ubuntu, usaremos el siguiente comando:

```
$ sudo apt install openjdk-11-jdk
```

A continuación mostramos como se ve este comando en la máquina virtual:

```
Adding debian:SSL.com_EV_Certification_Authority_RSA_R2.pem
Adding debian:DigiCert_Global_Root_CA.pem
Adding debian:Trustwave_Global_Certification_Authority.pem
Adding debian:D-TRUST_Root_Class_3_CA_2_2009.pem
Adding debian:Autoridad_de_Certificacion_FirmaProfesional_CIF_A62634068.pem
Adding debian:COMODO_Certification_Authority.pem
Adding debian:CFCA_EV_ROOT.pem
Adding debian:Entrust_Root_Certification_Authority_-_G4.pem
Adding debian:Trustwave_Global_ECC_P384_Certification_Authority.pem
Adding debian:SZAFIR_ROOT_CA2.pem
Adding debian:DigiCert_Assured_ID_Root_G2.pem
Adding debian:SwissSign_Gold_CA_-_G2.pem
Adding debian:Certigna_Root_CA.pem
Adding debian:Certum_Trusted_Network_CA.pem
Adding debian:Certum_Trusted_Network_CA_2.pem
Adding debian:Secure_Global_CA.pem
Adding debian:GLOBALTRUST_2020.pem
Adding debian:SSL.com_Root_Certification_Authority_ECC.pem
Adding debian:certSIGN_Root_CA_G2.pem
Adding debian:DigiCert_Assured_ID_Root_CA.pem
Adding debian:Staat_der_Nederlanden_EV_Root_CA.pem
Adding debian:OISTE_WISeKey_Global_Root_GC_CA.pem
Adding debian:GlobalSign_Root_CA_-_R3.pem
Adding debian:SSL.com_EV_Root_Certification_Authority_ECC.pem
Adding debian:Microsoft_ECC_Root_Certificate_Authority_2017.pem
Adding debian:AffirmTrust_Premium_ECC.pem
Adding debian:GlobalSign_Root_CA_-_R6.pem
Adding debian:emSign_Root_CA_-_C1.pem
Adding debian:Comodo AAA_Services_root.pem
Adding debian:Actalis_Authentication_Root_CA.pem
Adding debian:QuoVadis_Root_CA_2.pem
Adding debian:Starfield_Root_Certificate_Authority_-_G2.pem
Adding debian:SwissSign_Silver_CA_-_G2.pem
Adding debian:TWCA_Global_Root_CA.pem
Adding debian:Izenpe.com.pem
Adding debian:Starfield_Services_Root_Certificate_Authority_-_G2.pem
Adding debian:Atos_TrustedRoot_2011.pem
Adding debian:GlobalSign_Root_E46.pem
Adding debian:emSign_Root_CA_-_G1.pem
Adding debian:Certum_Trusted_Root_CA.pem
Adding debian:DigiCert_High_Assurance_EV_Root_CA.pem
Adding debian:IdenTrust_Commercial_Root_CA_1.pem
Adding debian:GlobalSign_ECC_Root_CA_-_R4.pem
Adding debian:T-TeleSec_GlobalRoot_Class_3.pem
Adding debian:certSIGN_ROOT_CA.pem
Adding debian:GTS_Root_R3.pem
Adding debian:Go_Daddy_Class_2_CA.pem
Adding debian:GlobalSign_Root_CA_-_R2.pem
Adding debian:Security_Communication_RootCA2.pem
Adding debian:Amazon_Root_CA_4.pem
Adding debian:UCA_Global_G2_Root.pem
Adding debian:DigiCert_Assured_ID_Root_G3.pem
done.
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
Processing triggers for systemd (245.4-4ubuntu3.21) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for ca-certificates (20211016ubuntu0.20.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

done.
done.
humbertowoodly@T5-2016630495-0:~$ █
```

Una vez instalado, verificaremos que contamos con la versión correcta disponible en nuestro sistema operativo:

```
done.  
humberto@T5-2016630495-0:~$ java --version  
openjdk 11.0.18 2023-01-17  
OpenJDK Runtime Environment (build 11.0.18+10-post-Ubuntu-0ubuntu120.04.1)  
OpenJDK 64-Bit Server VM (build 11.0.18+10-post-Ubuntu-0ubuntu120.04.1, mixed mode, sharing)  
humberto@T5-2016630495-0:~$ █
```

Copiando los archivos necesarios.

El último paso de configuración será copiar los archivos necesarios para la ejecución, estos archivos son:

- **MatrixClient.java**: es el archivo del Cliente, este sólamente será ejecutado en el Nodo 0 pero será copiado en todos porque es más fácil copiar \*.java.
- **MatrixOperations.java**: este es el archivo que contiene la interfaz de JavaRMI para realizar la distribución de la ejecución.
- **MatrixOperationsImpl.java**: este es el archivo con la implementación de la interfaz de JavaRMI.
- **MatrixServer.java**: este es el archivo con el código del servidor que se ejecutará en cada Nodo.

Para copiar los archivos, usaremos el comando [scp\(1\)](#), con el cual podemos copiar archivos desde y hacia nuestro servidor remoto usando SSH como protocolo de conectividad. El comando final será:

```
!ls  
MatrixClient.java MatrixOperations.java MatrixOperationsImpl.java MatrixServer.java README.md  
! scp ./azure/humbertowoody-azure.pem *.java humberto@172.198.15.187:/home/humbertowoody  
MatrixClient.java  
MatrixOperations.java  
MatrixOperationsImpl.java  
MatrixServer.java  
Warning: Identity file ./azure/humbertowoody-azure.pem not accessible: No such file or directory.  
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1037-azure x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
  
System information as of Mon May 1 19:18:21 UTC 2023  
System load: 0.0 Processes: 102  
Usage of /: 7.3% of 28.89GB Users logged in: 0  
Memory usage: 31% IPv4 address for eth0: 10.0.0.4  
Swap usage: 0%  
  
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s  
just raised the bar for easy, resilient and secure K8s cluster deployment.  
https://ubuntu.com/engage/secure-kubernetes-at-the-edge  
  
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  
  
New release '22.04.2 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last Login: Mon May 1 19:05:55 2023 from 189.132.192.41  
humberto@T5-2016630495-0:~$ ls  
MatrixClient.java MatrixOperations.java MatrixOperationsImpl.java MatrixServer.java  
humberto@T5-2016630495-0:~$ █
```

## Creación del resto de la topología

Los pasos que hemos seguido se repetirán en los otros dos nodos para completar la topología. Para esto, primero crearemos las otras dos máquinas virtuales requeridas, usaremos la misma configuración del Nodo 0 para autenticación y en tanto a recursos asignados:

Name	Type	Subscription	Resource group	Location	Status	Operating system	Size	Public IP address	Disk
TS-2016630495-0	Virtual machine	Azure for Students	SistemaDistribuidoResourceGroup	East US	Running	Linux	Standard_B1s	172.190.15.187	1
TS-2016630495-1	Virtual machine	Azure for Students	SistemaDistribuidoResourceGroup	East US	Running	Linux	Standard_B1s	74.235.232.33	1
TS-2016630495-2	Virtual machine	Azure for Students	SistemaDistribuidoResourceGroup	East US	Running	Linux	Standard_B1s	20.106.160.3	1

Lo siguiente será copiar los archivos en ambas máquinas virtuales, de la siguiente forma:

```

> scp -i ../../azur...@20.106.160.3:/home/humbertowood
The authenticity of host '20.106.160.3 (20.106.160.3)' can't be established.
ED25519 key fingerprint is SHA256:IN8Na0u8WQ21702/lxBeed8GDyKzSzAIZwSw2tHInU8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '20.106.160.3' (ED25519) to the list of known hosts.
MatrixClient.java                                100% 5192   72.3KB/s  00:00
MatrixOperations.java                            100% 1373   20.7KB/s  00:00
MatrixOperationsImpl.java                      100% 1000   15.6KB/s  00:00
MatrixServer.java                               100% 1028   13.1KB/s  00:00
> ssh -i ../../azur...@humbertowood...@20.106.160.3
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1037-azure x86_64)

 * Documentation: https://help.ubuntu.com
 * Management:   https://landscape.canonical.com
 * Support:      https://ubuntu.com/advantage

System information as of Mon May  1 19:31:45 UTC 2023

System load:  0.18    Processes:          102
Usage of /:  5.3% of 28.89GB  Users logged in:     0
Memory usage: 31%           IPv4 address for eth0: 10.0.0.6
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

New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

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

humbertowood...@TS-2016630495-2:~$ ls
MatrixClient.java MatrixOperations.java MatrixOperationsImpl.java MatrixServer.java
humbertowood...@TS-2016630495-2:~$ 

> scp -i ../../azur...@74.235.232.33:/home/humbertowood
The authenticity of host '74.235.232.33 (74.235.232.33)' can't be established.
ED25519 key fingerprint is SHA256:1MRXhrxAFF1iVVvSzYX6l+kz4WdTchvLMNjZ/314.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '74.235.232.33' (ED25519) to the list of known hosts.
MatrixClient.java                                100% 5192   83.0KB/s  00:00
MatrixOperations.java                            100% 193   3.1KB/s  00:00
MatrixOperationsImpl.java                      100% 1000   15.9KB/s  00:00
MatrixServer.java                               100% 1028   16.3KB/s  00:00
> ssh -i ../../azur...@74.235.232.33
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1037-azure x86_64)

 * Documentation: https://help.ubuntu.com
 * Management:   https://landscape.canonical.com
 * Support:      https://ubuntu.com/advantage

System information as of Mon May  1 19:30:35 UTC 2023

System load:  0.14    Processes:          101
Usage of /:  5.3% of 28.89GB  Users logged in:     0
Memory usage: 30%           IPv4 address for eth0: 10.0.0.5
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

New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

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

humbertowood...@TS-2016630495-1:~$ ls
MatrixClient.java MatrixOperations.java MatrixOperationsImpl.java MatrixServer.java
humbertowood...@TS-2016630495-1:~$ 

```

Y finalmente, instalar Java 11 en cada una y verificar su instalación:

```
Adding debian:Trustwave_Global_Certification_Authority.pem
Adding debian:Telia_Signing_Root_Class_3_G4_2_2009.pem
Adding debian:Autoridad_de_Certificación_Firmaprofesional_CIF_A62634068.pem
Adding debian:COMODO_Certification_Authority.pem
Adding debian:CFC_A_EV_ROOT.pem
Adding debian:Comodo_Ev_Certification_Authority_G4.pem
Adding debian:Trustwave_Global_ECC_P384_Certification_Authority.pem
Adding debian:SAFIR_ROOT_CA2.pem
Adding debian:SwissSign_Gold_CA_-G2.pem
Adding debian:SwissSign_Gold_CA_-G2.pem
Adding debian:Certigna_Root_CA.pem
Adding debian:Certum_Trusted_Network_CA.pem
Adding debian:GlobalSign_Root_CA_2.pem
Adding debian:Secure_Global_CA.pem
Adding debian:GLOBALTRUST_2020.pem
Adding debian:GlobalSign_ECC_Certification_Authority_ECC.pem
Adding debian:certSIGN_Root_CA_G2.pem
Adding debian:DigiCert_Assured_ID_Root_CA.pem
Adding debian:Staat_der_Nederlanden_EV_Root_CA.pem
Adding debian:GlobalSign_ECC_Certification_Authority_G2.pem
Adding debian:GlobalSign_Root_CA_-R3.pem
Adding debian:SSL_com_EV_Root_Certification_Authority_ECC.pem
Adding debian:GlobalSign_ECC_Certification_Authority_2017.pem
Adding debian:AffirmTrust_Premium_ECC.pem
Adding debian:GlobalSign_Root_CA_-R6.pem
Adding debian:eSign_Root_CA_G2.pem
Adding debian:GlobalSign_Services_root.pem
Adding debian:Actalis_Authentication_Root_CA.pem
Adding debian:QuVadis_Root_CA_2.pem
Adding debian:SwissSign_Silver_CA_-G2.pem
Adding debian:TWA_Global_Root_CA.pem
Adding debian:Izmpc_com.pem
Adding debian:GlobalSign_Root_Certificate_Authority_-G2.pem
Adding debian:Atos_TrustedRoot_2011.pem
Adding debian:GlobalSign_Root_E46.pem
Adding debian:Certigna_Trusted_Root_CA.pem
Adding debian:Certum_Trusted_Root_CA.pem
Adding debian:DigiCert_High_Assurance_EV_Root_CA.pem
Adding debian:IdenTrust_Commercial_Root_CA_1.pem
Adding debian:GlobalSign_Root_CA_-R6.pem
Adding debian:T-TelSec_GlobalRoot_Class_3.pem
Adding debian:certSIGN_ROOT_CA.pem
Adding debian:Go_Daddy_Class_2_CA.pem
Adding debian:GlobalSign_Root_CA_-R2.pem
Adding debian:Security_Communication_RootCA2.pem
Adding debian:UCA_Global_G2_Root.pem
Adding debian:DigiCert_Assured_ID_Root_G3.pem
done.
done.
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
Processing triggers for system (2.65-0ubuntu2.21) ...
Processing triggers for libssl1.0.2 (1.0.2-1) ...
Processing triggers for ca-certificates (20211016ubuntu20.04.1) ...
Updating certificates in /etc/ssl/certs...
# added, # removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
done.
humberto@woody:~$ 2016630495-2:~$ java --version
openjdk 11.0.18 2023-01-17
OpenJDK 64-Bit Server VM (build 11.0.18+18-post-Ubuntu-0ubuntu120.04.1, mixed mode, sharing)
humberto@woody:~$ 2016630495-2:~$
```

## Creación de Código (ChatGPT)

Para generar el código de esta tarea, se utilizó ChatGPT con la metodología que se empleó en la tarea anterior, primeramente se plantea la tarea en su totalidad y se verifica que el modelo comprenda los objetivos de la práctica, posteriormente se solicita uno a uno los programas a manera de maximizar la cantidad de tokens disponibles para cada respuesta y así permitir que el modelo pueda concluir en una iteración la generación del código necesario, finalmente se realizan preguntas generales para corregir detalles de implementación o posibles errores de ejecución.

A continuación, la conversación sostenida con ChatGPT:

Model: GPT-4

HU

holá! necesito realizar la siguiente tarea, abajo colocaré la descripción:



Utilizando ChatGPT se deberá desarrollar un sistema que calcule el producto de dos matrices rectangulares utilizando Java RMI, tal como se explicó en clase.

Sea la matriz A con dimensiones NxM, y la matriz B con dimensiones MxN.

Sea  $C = A \times B$ .

Se deberán ejecutar dos casos:

$N=9, M=4$  se deberá desplegar la matriz C y el checksum de la matriz C.

$N=900, M=400$  se deberá desplegar el checksum de la matriz C.

Los elementos de las matrices A, B y C deberán ser de tipo float y el checksum deberá ser de tipo double.

Se deberá inicializar las matrices A y B de la siguiente manera:

$$A[i][j] = 2 * i + 3 * j$$

$$B[i][j] = 3 * i - 2 * j$$

Sea BT la transpuesta de la matriz B.

Se deberá dividir las matrices A y BT en nueve partes iguales, por tanto la matriz C estará dividida en 81 partes:

La multiplicación de las matrices  $A_i \times B_{Tj}$  deberá realizarse renglón por renglón, tal como vimos en la clase "Jerarquía de memoria".

Se deberá crear tres máquinas virtuales con Ubuntu en el mismo grupo de recursos, de acuerdo a la siguiente topología:

La máquina virtual 0 (nodo 0) ejecutará el cliente RMI el cual hará los siguientes pasos:

Inicializar las matrices A y B.

Obtener la matriz BT.

Obtener las matrices  $A_i$  ( $i=1,2,\dots,9$ ) y  $B_{Ti}$  ( $i=1,2,\dots,9$ )

Utilizando RMI obtener las matrices C1, C2, ... C27 invocando el método `multiplica_matrices()` el cual ejecutará en el nodo 0.

Utilizando RMI obtener las matrices C28, C29, ... C54 invocando el método `multiplica_matrices()` el cual ejecutará en el nodo 1.

Utilizando RMI obtener las matrices C55, C56, ... C81 invocando el método `multiplica_matrices()` el cual ejecutará en el nodo 2.

Obtener la matriz C a partir de las matrices C1, C2, ..., C81

Para el caso  $N=9$  y  $M=4$  desplegar

Regenerate response

Calcular y desplegar el checksum de la matriz C

Send a message.



Para el caso  $N=9$  y  $M=4$  desplegar la matriz C.  
Calcular y desplegar el checksum de la matriz C.  
Los nodos 0, 1 y 2 ejecutarán el servidor RMI y el rmiregistry.

El cliente RMI deberá utilizar threads para invocar el método remoto multiplica\_matrices() de manera que los servidores RMI calculen los productos en paralelo.  
Se deberá subir a la plataforma el código fuente del sistema desarrollado (la interface, la clase, el servidor y el cliente) y un reporte de la tarea en formato PDF con portada, desarrollo y conclusiones.

El reporte PDF deberá incluir las capturas de pantalla de la compilación y ejecución del programa, se deberá incluir la captura de pantalla correspondiente a cada paso de la creación de la máquina virtual 0.

No se admitirá la tarea si no incluye las pantallas correspondientes a cada paso del procedimiento de creación de la máquina virtual 0.

El reporte PDF deberá incluir la conversación realizada con ChatGPT.

El nombre de cada máquina virtual deberá ser: "T5-" concatenando el número de boleta del alumno o alumna, un guion y el número de máquina virtual, por ejemplo, si el número de boleta es 12345678, entonces la máquina virtual 0 deberá llamarse: T5-12345678-0, la máquina virtual 1 deberá llamarse T5-12345678-1, y así sucesivamente. No se admitirá la tarea si las máquinas virtuales no se nombran como se indicó anteriormente.

Los checksum deben ser correctos.

¿Podrías darme el código en java para la clase y la interfaz? luego te pediré el código del cliente y del servidor, gracias!



Claro, a continuación, te muestro el código en Java para la interfaz y la clase. Comencemos con la interfaz:

MatrixOperations.java:

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

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

Ahora, la clase que implementa la interfaz:

MatrixOperationsImpl.java:

Regenerate response

Send a message.



ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT Mar 23 Version](#)

Ahora, la clase que implementa la interfaz:

MatrixOperationsImpl.java:

```
java Copy code

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

public class MatrixOperationsImpl extends UnicastRemoteObject implements Mat

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

    @Override
    public float[][] multiplyMatrices(float[][] A, float[][] B) throws RemoteException {
        int aRows = A.length;
        int aColumns = A[0].length;
        int bRows = B.length;
        int bColumns = B[0].length;

        if (aColumns != bRows) {
            throw new IllegalArgumentException("A:Filas: " + aColumns + " no "
        }

        float[][] C = new float[aRows][bColumns];
        for (int i = 0; i < aRows; i++) {
            for (int j = 0; j < bColumns; j++) {
                C[i][j] = 0.0f;
                for (int k = 0; k < aColumns; k++) {
                    C[i][j] += A[i][k] * B[k][j];
                }
            }
        }

        return C;
    }
}
```

Ahora que tienes el código de la interfaz y la clase, si necesitas ayuda con el código del cliente y del servidor, no dudes en pedírmelo.

HU excelente! ahora podrías darme el código del cliente porfavor? gracias!

Claro, aquí está el código del cliente:

Regenerate response

MatrixClient.java:

Send a message.

ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT Mar 23 Version](#)

Claro, aquí está el código del cliente:

MatrixClient.java:

```
java
import java.rmi.Naming;
import java.rmi.RemoteException;
import java.util.ArrayList;
import java.util.List;

public class MatrixClient {

    public static void main(String[] args) {
        try {
            String[] nodes = {"rmi://localhost/node0", "rmi://localhost/node1", "rmi://localhost/node2", "rmi://localhost/node3", "rmi://localhost/node4", "rmi://localhost/node5", "rmi://localhost/node6", "rmi://localhost/node7", "rmi://localhost/node8"};

            // Inicializa las matrices A y B
            int N = 9;
            int M = 4;
            float[][] A = initializeMatrixA(N, M);
            float[][] B = initializeMatrixB(M, N);
            float[][] BT = transposeMatrix(B);

            // Divide las matrices A y BT en 9 partes iguales
            List<float[][]> A_parts = divideMatrix(A, 3);
            List<float[][]> BT_parts = divideMatrix(BT, 3);

            // Realiza las multiplicaciones utilizando RMI y threads
            float[][][] C_parts = new float[81][][][];
            List<Thread> threads = new ArrayList<>();
            for (int i = 0; i < 9; i++) {
                for (int j = 0; j < 9; j++) {
                    int index = i * 9 + j;
                    int nodeIndex = index / 27;
                    MatrixOperations matrixOperations = (MatrixOperations) Naming.lookup("rmi://" + nodes[nodeIndex]);
                    Runnable task = () -> {
                        try {
                            C_parts[index] = matrixOperations.multiplyMatrix(A_parts.get(i), BT_parts.get(j));
                        } catch (RemoteException e) {
                            e.printStackTrace();
                        }
                    };
                    Thread thread = new Thread(task);
                    threads.add(thread);
                    thread.start();
                }
            }

            // Espera a que todos los hilos terminen
            for (Thread thread : threads) {
                thread.join();
            }
        }
    }
}
```

Send a message.

ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT Mar 23 Version](#)

```
}

    // Espera a que todos los threads terminen
    for (Thread thread : threads) {
        thread.join();
    }

    // Combina las matrices C_parts en la matriz C
    float[][] C = combineMatrix(C_parts, 3, N);

    // Despliega la matriz C y su checksum
    if (N == 9 && M == 4) {
        printMatrix(C);
    }
    double checksum = calculateChecksum(C);
    System.out.println("Checksum: " + checksum);

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

// Métodos auxiliares para inicializar matrices, dividir, transponer, co
}
```

Asegúrate de agregar los métodos auxiliares para inicializar las matrices, dividir, transponer, combinar y calcular el checksum que no están incluidos en el código proporcionado. Si necesitas ayuda con esos métodos, avísame y te los proporcionaré. No olvides cambiar las direcciones de los nodos en el array `nodes` según tu configuración.

HU perfecto! haré los cambios en el array `nodes` como sugieres. podrías darme ahora el código del servidor?

Claro, aquí tienes el código del servidor:



MatrixServer.java:

```
java
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;

public class MatrixServer {

    public static void main(String[] args) {
        if (args.length != 1) {
            Regenerate response
        }
    }
}
```

Send a message.



ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT Mar 23 Version](#)

HU perfecto! haré los cambios en el array `nodes` como sugieres. podrías darme ahora el código del servidor?



Claro, aquí tienes el código del servidor:



MatrixServer.java:

```
java Copy code  
  
import java.rmi.registry.LocateRegistry;  
import java.rmi.registry.Registry;  
import java.rmi.server.UnicastRemoteObject;  
  
public class MatrixServer {  
  
    public static void main(String[] args) {  
        if (args.length != 1) {  
            System.out.println("Uso: java MatrixServer <node_id>");  
            System.exit(1);  
        }  
  
        int nodeId = Integer.parseInt(args[0]);  
  
        try {  
            // Crea e instancia el objeto remoto  
            MatrixOperationsImpl obj = new MatrixOperationsImpl();  
            MatrixOperations stub = (MatrixOperations) UnicastRemoteObject.e  
  
            // Registra el objeto remoto en el rmiregistry  
            Registry registry = LocateRegistry.getRegistry();  
            registry.rebind("node" + nodeId, stub);  
  
            System.out.println("Servidor RMI (nodo " + nodeId + ") listo...")  
        } catch (Exception e) {  
            System.err.println("Error en el servidor (nodo " + nodeId + "):");  
            e.printStackTrace();  
        }  
    }  
}
```

Antes de ejecutar el servidor, asegúrate de iniciar el `rmiregistry` en cada uno de los nodos. También, debes proporcionar el ID del nodo como argumento al ejecutar el servidor (e.g., `java MatrixServer 0` para el nodo 0).

Esto debería ser suficiente para implementar tu sistema. Si necesitas ayuda adicional, no dudes en preguntar. ¡Buena suerte! ⟳ Regenerate response

Send a message.



HU veo que faltaron algunos métodos en el cliente! podrías darme el código para inicializar matrices, dividir y transponer?

Mis disculpas por no proporcionar esos métodos anteriormente. Aquí tienes los métodos auxiliares para inicializar matrices, dividir y transponer:

```
java
private static float[][] initializeMatrixA(int N, int M) {
    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;
        }
    }
    return A;
}

private static float[][] initializeMatrixB(int M, int N) {
    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;
        }
    }
    return B;
}

private static float[][] transposeMatrix(float[][] matrix) {
    int rows = matrix.length;
    int columns = matrix[0].length;
    float[][] transposed = new float[columns][rows];
    for (int i = 0; i < columns; i++) {
        for (int j = 0; j < rows; j++) {
            transposed[i][j] = matrix[j][i];
        }
    }
    return transposed;
}

private static List<float[][]> divideMatrix(float[][] matrix, int parts) {
    List<float[][]> dividedMatrix = new ArrayList<>();
    int rows = matrix.length;
    int columns = matrix[0].length;
    int blockRows = rows / parts;
    int blockColumns = columns / parts;

    for (int i = 0; i < par
    Regenerate response

```

Send a message. 🔗

ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT Mar 23 Version](#)

```
private static List<float[][]> divideMatrix(float[][] matrix, int parts) {
    List<float[][]> dividedMatrix = new ArrayList<>();
    int rows = matrix.length;
    int columns = matrix[0].length;
    int blockRows = rows / parts;
    int blockColumns = columns / parts;

    for (int i = 0; i < parts; i++) {
        for (int j = 0; j < parts; j++) {
            float[][] block = new float[blockRows][blockColumns];
            for (int k = 0; k < blockRows; k++) {
                System.arraycopy(matrix[i * blockRows + k], j * blockColumns
            }
            dividedMatrix.add(block);
        }
    }
    return dividedMatrix;
}
```

Agrega estos métodos en la clase MatrixClient.java donde se menciona "Métodos auxiliares para inicializar matrices, dividir, transponer, combinar y calcular el checksum aquí". Estos métodos te permitirán inicializar las matrices A y B, transponer la matriz B y dividir las matrices A y BT en partes iguales como se describe en la tarea.



genial! ahora me podrías dar los métodos para combinar y calcular el checksum? gracias!



Por supuesto, aquí tienes los métodos para combinar las matrices y calcular el checksum:



```
java Copy code

private static float[][] combineMatrix(float[][][] C_parts, int parts, int N) {
    int blockRows = N / parts;
    int blockColumns = N / parts;
    float[][] C = new float[N][N];

    for (int i = 0; i < parts; i++) {
        for (int j = 0; j < parts; j++) {
            float[][] block = C_parts[i * parts + j];
            for (int k = 0; k < blockRows; k++) {
                System.arraycopy(block[k], 0, C[i * blockRows + k], j * blockColumns
            }
        }
    }

    return C;
}
```

Send a message. Regenerate response

ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT Mar 23 Version](#)



Por supuesto, aquí tienes los métodos para combinar las matrices y calcular el checksum:



Copy code

```
java

private static float[][] combineMatrix(float[][][] C_parts, int parts, int N) {
    int blockRows = N / parts;
    int blockColumns = N / parts;
    float[][] C = new float[N][N];

    for (int i = 0; i < parts; i++) {
        for (int j = 0; j < parts; j++) {
            float[][] block = C_parts[i * parts + j];
            for (int k = 0; k < blockRows; k++) {
                System.arraycopy(block[k], 0, C[i * blockRows + k], j * blockColumns, blockRows);
            }
        }
    }

    return C;
}

private static double calculateChecksum(float[][] matrix) {
    double checksum = 0;
    for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix[0].length; j++) {
            checksum += matrix[i][j];
        }
    }
    return checksum;
}

private static void printMatrix(float[][] matrix) {
    for (float[] row : matrix) {
        for (float value : row) {
            System.out.printf("%.2f ", value);
        }
        System.out.println();
    }
}
```

Agrega estos métodos a la clase MatrixClient.java donde se menciona "Métodos auxiliares para inicializar matrices, dividir, transponer, combinar y calcular el checksum aquí". Estos métodos te permitirán combinar las matrices C\_parts en la matriz C y calcular el checksum de la matriz C.

También incluí un método adicional `printMatrix` para imprimir la matriz C en caso de que deseas visualizarla en la consola. Puedes agregarlo al código del cliente en la misma sección que los otros métodos auxiliares.

Regenerate response

Send a message.



ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT Mar 23 Version](#)

## Ejecución del Programa

Para ejecutar este programa, primero deberemos iniciar el Servidor en los tres nodos, así como el Registry de JavaRMI para que puedan invocarse métodos remotos. Posteriormente, ejecutaremos el cliente en el Nodo 0 y nos aseguraremos que el checksum mostrado sea el correcto. Para las pruebas a realizar, usará la siguiente distribución de ventanas:

- Izquierda:
  - Arriba: será el Nodo 0, el cual ejecuta un Servidor y un Registry.
  - Abajo: será el mismo Nodo 0, pero aquí ejecutaremos el Cliente.
- Derecha:
  - Arriba: será el Nodo 2, el cual ejecuta un Servidor y un Registry.
  - Abajo: será el Nodo 1, el cual ejecuta un Servidor y un Registry.

Las modificaciones realizadas en el código para ejecutarlo en la topología en la nube son:

- Cambiar localhost por las IP privadas (dentro de la VPC de Azure) de cada uno de los nodos correspondientes.
- Los valores de **N** y **M** serán cambiados en el cliente para cumplir con los casos planteados.
  - Para el caso dónde N=9 y M=4 además se añadió la llamada a la función printMatrix() con el fin de mostrar la matriz final.

La compilación del código, registro de la interfaz y ejecución de un nodo se ve de la siguiente forma:

```
> javac *.java
> rmic MatrixOperationsImpl
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
> rmiregistry &
[1] 34576
> java MatrixServer 0
Servidor RMI (nodo 0) listo...
```

Para las ejecuciones finales se asumirá estos mismos comandos y ciclo de ejecución y, dado que el proceso del Registry de RMI se envía al *background*, no veremos su salida en stdout.

*Nota: probablemente exista una forma más elegante de ejecutar todo al mismo tiempo pero para fines prácticos cumple la funcionalidad esperada de la práctica.*

## Primer caso

El primer caso a probar es dónde  $N=9$  y  $M=4$ , el resultado es el siguiente:

```

humbertowooddy@T5-2016630495-0:~$ rmiregistry &
[3] 9995
[2]  Exit 1          rmiregistry
humbertowooddy@T5-2016630495-0:~$ java rmi.server.ExportException: Port already in use: 1099; nested exception is:
java.net.BindException: Address already in use (Bind failed)
at java.rmi.sun.rmi.transport.tcp.TCPEndpoint.listen(TCPEndpoint.java:335)
at java.rmi.sun.rmi.transport.tcp.TCPEndpoint.exportObject(TCPEndpoint.java:41
2)
at java.rmi.sun.rmi.transport.tcp.TCPEndpoint.exportObject(TCPEndpoint.java:41
2)
at java.rmi.sun.rmi.transport.LiveRef.exportObject(LiveRef.java:147)
at java.rmi.sun.rmi.server.UnicastServerRef.exportObject(UnicastServerRef.java
234)
at java.rmi.sun.rmi.registry.RegistryImpl.setup(RegistryImpl.java:220)
at java.rmi.sun.rmi.registry.RegistryImpl$2.run(RegistryImpl.java:196)
at java.rmi.sun.rmi.registry.RegistryImpl$2.run(RegistryImpl.java:193)
at java.base/java.security.AccessController.doPrivileged(Native Method)
at java.base/java.security.AccessController.doPrivileged(AccessController.java
726)
at java.rmi.sun.rmi.registry.RegistryImpl.main(RegistryImpl.java:193)
at java.rmi.sun.rmi.registry.RegistryImpl$5.run(RegistryImpl.java:531)
at java.rmi.sun.rmi.registry.RegistryImpl$5.run(RegistryImpl.java:529)
at java.base/java.security.AccessController.doPrivileged(Native Method)
at java.rmi.sun.rmi.registry.RegistryImpl.createRegistry(RegistryImpl.java:528
)
Caused by: java.net.BindException: Address already in use (Bind failed)
at java.base/java.net.BindException.createSocketImpl(SocketImpl$Native Method)
at java.base/java.net.AbstractPlainSocketImpl.bind(AbstractPlainSocketImpl.java
a:452)
at java.base/java.net.ServerSocket.bind(ServerSocket.java:395)
at java.base/java.net.ServerSocket.<init>(ServerSocket.java:257)
at java.base/java.net.ServerSocket.<init>(ServerSocket.java:149)
at java.rmi.sun.rmi.transport.tcp.TCPDirectSocketFactory.createServerSocket(TC
PDIRECTSocketFactory.java:45)
at java.rmi.sun.rmi.transport.tcp.TCPEndpoint.newServerSocket(TCPEndpoint.java
:670)
at java.rmi.sun.rmi.transport.tcp.TCPEndpoint.listen(TCPEndpoint.java:324)
... 15 more
^C
[3]  Exit 1          rmiregistry
humbertowooddy@T5-2016630495-0:~$ java MatrixServer 0
Servidor RMI (nodo 0) listo...
|
```

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  
-1053.0  
humbertowooddy@T5-2016630495-0:~\$

El resultado obtenido (checksum) es: -10530.0

Nota: Tuve un problema con el registry porque, al parecer, falló en su ejecución en segundo plano y luego no permitía reusar el mismo puerto así que tuve que conectarme por otra sesión de SSH, matar el proceso que tenía el puerto abierto y reiniciar el registry y el servidor en el Nodo 0.

## Segundo caso

El segundo caso a probar es dónde  $N=900$  y  $M=400$ , el resultado es el siguiente:

```
humbertowoodytS-2016630495-0:-$ javac *.java
humbertowoodytS-2016630495-0:-$ ls
MatrixClient.class MatrixOperations.java MatrixOperationsImpl_Stub.class
MatrixClient.java MatrixOperationsImpl.class MatrixServer.class
MatrixOperations.class MatrixOperationsImpl.java MatrixServer.java
humbertowoodytS-2016630495-0:-$ rmic MatrixOperationsImpl
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
humbertowoodytS-2016630495-0:-$ rmiregistry 6
[1] 946
humbertowoodytS-2016630495-0:-$ java MatrixServer 0
Servidor RMI (nodo 0) listo...

humbertowoodytS-2016630495-0:-$ java MatrixClient
-1.06919889581881E14
humbertowoodytS-2016630495-0:-$ [ ]
```

```
humbertowoodytS-2016630495-2:-$ javac *.java
humbertowoodytS-2016630495-2:-$ ls
MatrixClient.class MatrixOperations.class MatrixOperationsImpl.class MatrixServer.class
MatrixClient.java MatrixOperations.java MatrixOperationsImpl.java MatrixServer.java
humbertowoodytS-2016630495-2:-$ rmic MatrixOperationsImpl
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
humbertowoodytS-2016630495-2:-$ rmiregistry 6
[1] 10865
humbertowoodytS-2016630495-2:-$ java MatrixServer 2
Servidor RMI (nodo 2) listo...
```

```
humbertowoodytS-2016630495-1:-$ javac *.java
humbertowoodytS-2016630495-1:-$ ls
MatrixClient.class MatrixOperations.class MatrixOperationsImpl.class MatrixServer.class
MatrixClient.java MatrixOperations.java MatrixOperationsImpl.java MatrixServer.java
humbertowoodytS-2016630495-1:-$ rmic MatrixOperationsImpl
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
humbertowoodytS-2016630495-1:-$ rmiregistry 6
[1] 11538
humbertowoodytS-2016630495-1:-$ java MatrixServer 1
Servidor RMI (nodo 1) listo...
```

El resultado obtenido (checksum): -1.06919889581881E14

## Conclusión

En la realización de esta tarea aprendí sobre el uso de JavaRMI para poder distribuir a nivel código la ejecución de un programa entre múltiples nodos. El poder realizarlo de esta forma optimiza bastante la complejidad de ejecución y nos permite sacar provecho del hardware disponible en cada uno de los nodos así como del escalamiento horizontal que nos permite la topología planteada.

Me pareció un poco complicado lograr que ChatGPT diera al principio el código y tuve que realizar pequeñas adecuaciones al código generado puesto que algunas cosas (índices en su gran mayoría) estaban invertidos o simplemente en un orden incorrecto.

Fue un problema mayúsculo arreglar la función de dividir matrices, por algún motivo ChatGPT simplemente no quería cooperar con la causa. Hubo que hacer una intervención manual porque ya estaba arrojando resultados muy desquiciados y absurdamente irracionales.

– Humberto Alejandro Ortega Alcocer.