



Instituto Politécnico Nacional

Escuela Superior de Computo



Tarea 3. Multiplicación de matrices distribuida utilizando paso de mensajes.

Alumno: Javier Martínez Carranza

Materia: Desarrollo de Sistemas Distribuidos

Grupo: 4CV11

Profesor: Carlos Pineda Guerrero

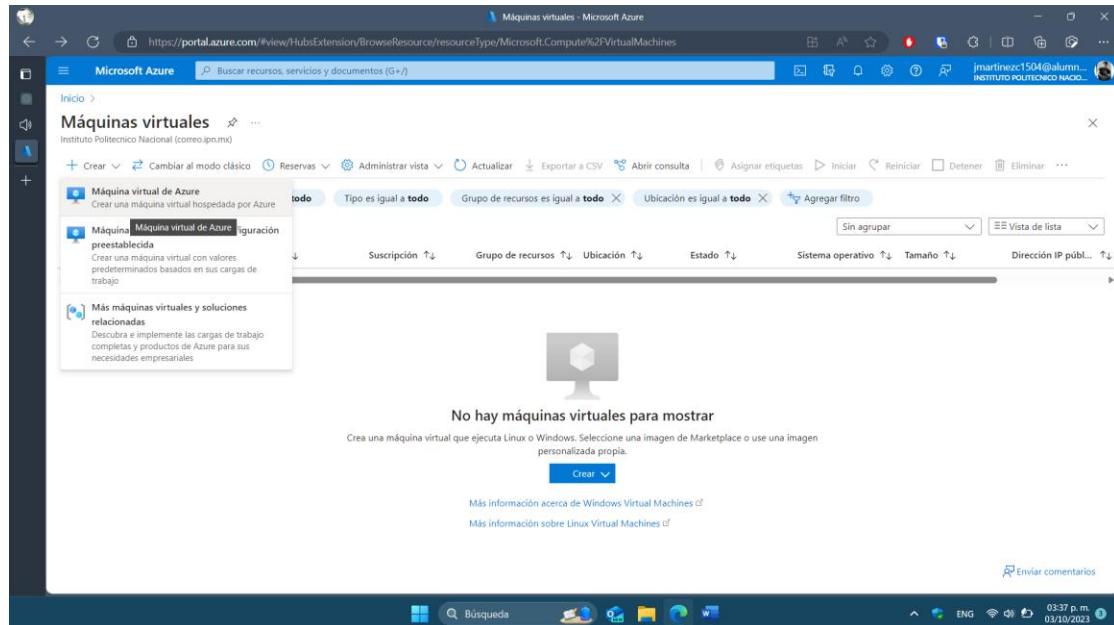
índice

Desarrollo	1
Creación de maquina virtual para el nodo 1	1
Conversaciones con las IAs.	8
Códigos	25
Ejecución y pruebas	30
Prueba 1. Matriz de 6x5.....	31
Prueba 2. Matriz de 3000x2000.....	34
Conclusión	35

Desarrollo

Creación de maquina virtual para el nodo 1

En este paso veremos la creación de la máquina virtual para el nodo 1.



Seleccionamos algún grupo de recursos que tengamos creado, o en su defecto creamos uno nuevo, de igual forma ponemos el nombre como viene indicado en la tarea.

Dejamos la zona por default o en su defecto seleccionamos la que nos cueste más barato, esta máquina funcionará con Ubuntu 20.04 con un tamaño de 1vcpu 1GiB de memoria, el precio por esto será de 7.59\$/mes.

Zona de disponibilidad * Ahora puede seleccionar varias zonas. Si selecciona varias zonas, se creará una VM por zona. [Más información](#)

Tipo de seguridad * [Configurar características de seguridad](#)

Imagen * [Ver todas las imágenes](#) | [Configurar la generación de máquinas virtuales](#)

Arquitectura de VM * Arm64 x64

Ejecución de Azure Spot con descuento

Tamaño * [Ver todos los tamaños](#)

Cuenta de administrador

Tipo de autenticación *

[Revisar y crear](#) [Anterior](#) [Siguiente: Discos >](#) [Enviar comentarios](#)

En el tipo de autenticación seleccionaremos contraseña y crearemos un usuario y contraseña para este usuario, de igual forma dejamos por default el puerto SSH(22) con el cual accederemos a la máquina, después de este paso daremos en “Discos”.

Cuenta de administrador

Tipo de autenticación *

Nombre de usuario *

Contraseña *

Confirmar contraseña *

Reglas de puerto de entrada

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

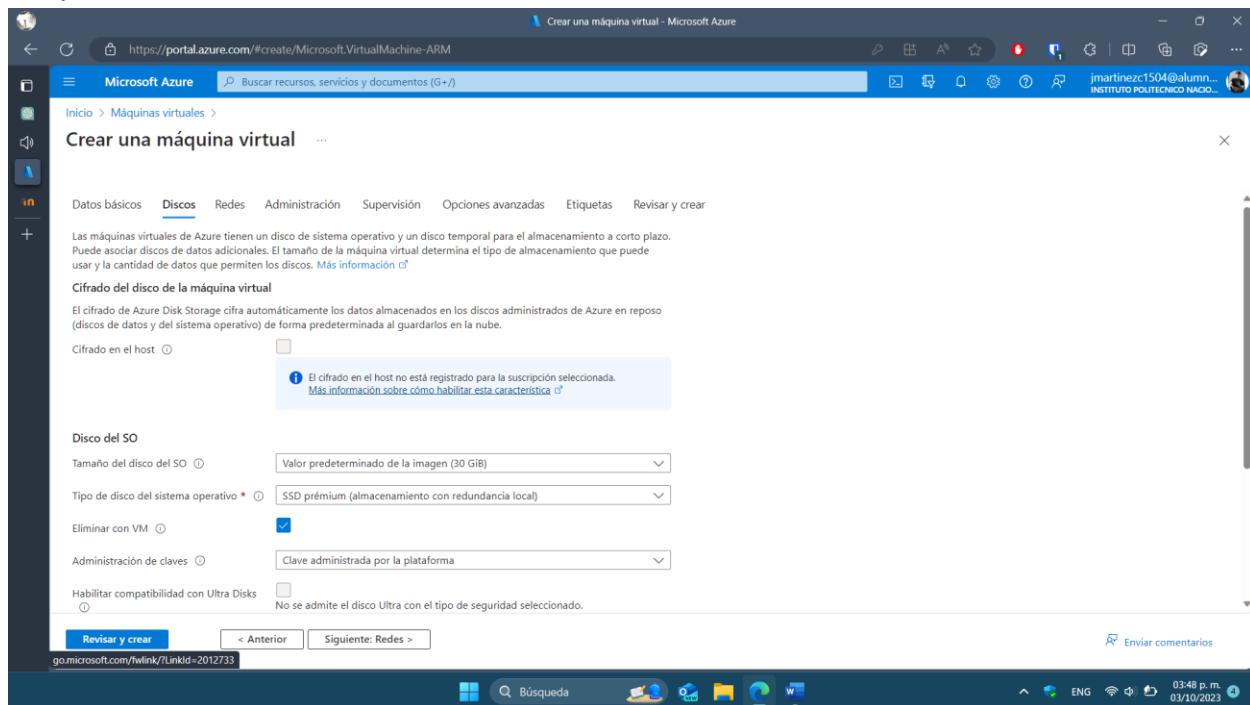
Puertos de entrada públicos * Ninguno Permitir los puertos seleccionados

Seleccionar puertos de entrada *

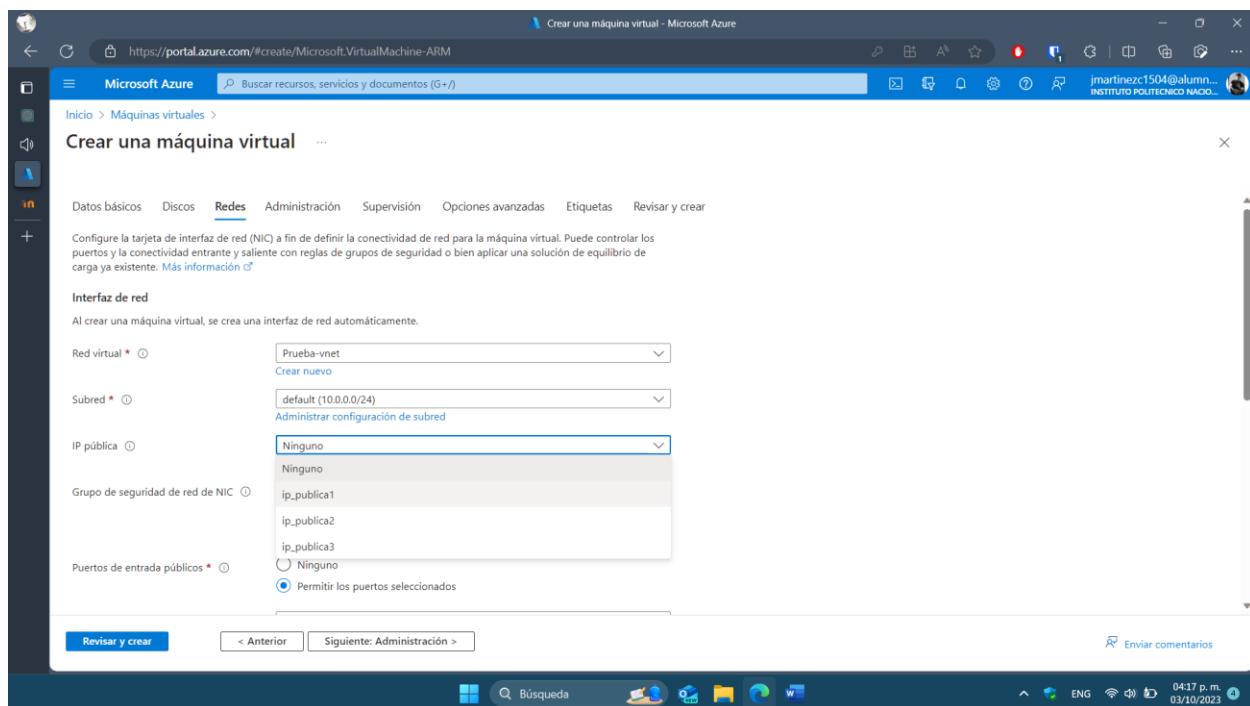
Se bloquea todo el tráfico de internet de forma predeterminada. Puede cambiar las reglas del puerto de entrada en la página VM > Redes.

[Revisar y crear](#) [Anterior](#) [Siguiente: Discos >](#) [Enviar comentarios](#)

En el apartado de discos dejaremos el tamaño predeterminado de 30GiB y pasaremos al apartado de “Redes”.



En el apartado de Redes/”IP pública” seleccionamos alguna de las 3 disponibles que Azure nos proporciona para usar, en este caso seleccione la primera. Hecho “Siguiente: Administración” y luego “Siguiente: Supervisión”.



En este apartado simplemente deshabilitamos la opción de diagnostico y hecho esto en este punto ya podemos dar en “Revisar y Crear”.

The screenshot shows the 'Crear una máquina virtual' (Create a virtual machine) wizard in the Microsoft Azure portal. The 'Supervisión' (Monitoring) tab is active. Under 'Diagnóstico', the 'Deshabilitar' (Disable) radio button is selected for both 'Diagnósticos de arranque' (Boot Diagnostics) and 'Habilitar diagnósticos del SO invitado' (Enable guest OS diagnostics). At the bottom, there are buttons for 'Revisar y crear' (Review + Create) and 'Siguiente: Opciones avanzadas' (Next: Advanced options).

Esperamos un momento y si todo es correcto, mostrara la siguiente pantalla, verificamos la información de nuestra máquina virtual y si es correcto, damos en “Crear”

The screenshot shows the final review step of the 'Crear una máquina virtual' (Create a virtual machine) wizard. The 'Revisar y crear' (Review + Create) tab is active. A green bar at the top indicates 'Validación superada' (Validation passed). Below, it shows the price for '1 X Standard B1s by Microsoft' at '0,0104 USD/hr'. There is a note: '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.' (The cost indicated below is an estimate and not the final price. Use the [Price calculator](#) for all your needs.) The 'TERMS' section contains legal text. A 'Name' field is filled with 'JAVIER MARTINEZ CARRANZA'. At the bottom, there are buttons for 'Crear' (Create) and 'Siguiente >' (Next >).

Después de esperar un tiempo, en nuestras通知 se mostrarán las opciones para ir al recurso o anclar el panel, en este caso iremos al recurso que es nuestra máquina virtual.

The screenshot shows the Microsoft Azure portal interface. At the top, it displays the URL <https://portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2fsubscriptions%2f954bd92c-20f2-4909-bee0-98d8f5...>. The main content area is titled "CreateVm-canonical.0001-com-ubuntu-server-focal-2-20231003153741 | Información general". A green checkmark indicates "Se completó la implementación" (Implementation completed). Below this, there are sections for "Nombre de implementación", "Hora de inicio", "Suscripción", and "Grupo de recursos". On the right side, there are promotional cards for "Cost Management", "Microsoft Defender for Cloud", and "Trabajar con un experto". At the bottom, there are buttons for "Ir al recurso" and "Anclar al panel".

Desde aquí podremos iniciar, detener o reiniciar nuestra máquina virtual.

The screenshot shows the Microsoft Azure portal interface for a specific VM named "T3-2019630039-1". The left sidebar lists navigation options such as "Información general", "Control de acceso (IAM)", "Etiquetas", "Diagnosticar y solucionar problemas", "Conectar", "Redes", "Configuración", and "Discos". The main content area displays "Información esencial" with details like Grupo de recursos (mover), Estado (En ejecución), Ubicación (East US (Zona 1)), Suscripción (mover), Id. de suscripción, Zona de disponibilidad, and Etiquetas (editar). Below this, there are tabs for "Propiedades", "Supervisión", "Funcionalidades (7)", "Recomendaciones", and "Tutorial". On the right, there are sections for "Máquina virtual" (with fields like Nombre del equipo, Sistema operativo, Editor de imágenes, Oferta de imagen, Plan de imagen, Generación de VM, Arquitectura de VM, and Estado del agente) and "Redes" (with fields like Dirección IP pública, Dirección IP privada, Red virtual/subred, and Nombre DNS). The bottom of the screen shows the Windows taskbar and system tray.

Antes de terminar con nuestra creación de la máquina virtual, abriremos un puerto en el apartado Redes/Configuración de red.

The screenshot shows the Microsoft Azure portal interface. The left sidebar has 'Redes' (Network) selected under 'Configuración'. The main area shows the 'T3-2019630039-1 | Configuración de red' page. In the 'Reglas' (Rules) section, there is a table listing rules for an NSG. A context menu is open over the 'Regla de puerto de entrada' (Inbound port rule) option, with options like '+ Crear ACL del puerto' (Create port ACL), 'Regla de puerto de entrada' (Inbound port rule), and 'Regla de puerto de salida' (Outbound port rule).

Estando en la configuración, abriremos el puerto 50000 y con el protocolo TCP.

The screenshot shows the Microsoft Azure portal interface. The left sidebar has 'Redes' (Network) selected under 'Configuración'. The main area shows the 'T3-2019630039-1 | Configuración de red' page. A modal dialog box titled 'Agregar regla de seguridad de entrada' (Add inbound security rule) is open. It shows fields for 'Origen' (Source) set to 'Any', 'Intervalos de puertos de origen' (Source port range) set to '*', 'Destino' (Destination) set to 'Any', 'Servicio' (Service) set to 'Custom', 'Intervalos de puertos de destino' (Destination port range) set to '50000', 'Protocolo' (Protocol) set to 'TCP' (selected), and 'Acción' (Action) set to 'Permitir' (Allow). At the bottom of the dialog are 'Agregar' (Add) and 'Cancelar' (Cancel) buttons.

Una vez abierto los puertos ya quedará completado el proceso de creación de nuestra máquina virtual del nodo 1.

The screenshot shows the Microsoft Azure portal interface for a virtual machine named T3-2019630039-1. The left sidebar navigation includes 'Máquina virtual', 'Conectar', 'Redes' (selected), and 'Configuración'. The main content area is titled 'T3-2019630039-1 | Configuración de red'. It displays the 'Reglas' (Rules) section for a Network Security Group (NSG). A message at the top says 'Esta es una experiencia den versión preliminar. Proporciona comentarios'. The table lists five rules:

Prioridad	Nombre	Puerto	Protocolo	Origen	Destino
300	SSH	22	TCP	Cualquiera	Cualquiera
310	Puerto50000	50000	TCP	Cualquiera	Cualquiera
65000	AllowVnetInBound	Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork
65001	AllowAzureLoadBalancerInBound	Cualquiera	Cualquiera	AzureLoadBalancer	Cualquiera
65500	DenyAllInBound	Cualquiera	Cualquiera	Cualquiera	Cualquiera

Conversaciones con las IAs.

En esta tarea usamos la IA de Microsoft Bing. Primero le pasamos los parámetros a Bing AI como se muestra a continuación.

The screenshot shows a Microsoft Bing Chat window. The main area displays a text block from the AI:

En esta tarea se deberá desarrollar un solo programa en Java el cual calculará el producto de dos matrices rectangulares en forma distribuida utilizando tres máquinas virtuales de Azure. Sean A y B matrices rectangulares con elementos de tipo double, la dimensión de la matriz A es N renglones y M columnas, y la dimensión de la matriz B es M renglones y N columnas. Sea C=AxB matriz cuadrada con N renglones y N columnas. El checksum de la matriz C se calculará como la suma de todos elementos de la matriz. El checksum deberá ser de tipo double.

Se deberá inicializar las matrices de la siguiente manera:
A[i][j] = 5*i+3*j
B[i][j] = 6*i+3*j
Donde A[i][j] y B[i][j] son los elementos A_{i,j} y B_{i,j} respectivamente.

¿Cómo realizar la multiplicación de matrices en forma distribuida? Sea BT la transpuesta de la matriz B. Se deberá dividir las matrices A y BT en seis partes de igual tamaño, por tanto la matriz C estará dividida en 36 partes como se muestra:
A=[A1,A2,A3,A4,A5,A6]
BT=[BT1,BT2,BT3,BT4,BT5,BT6]
C=[[C1,C2,C3,C4,C5,C6],[C7,C8,C9,C10,C11,C12],[C13,C14,C15,C16,C17,C18],[C19,C20,C21,C22,C23,C24],[C25,C26,C27,C28,C29,C30],[C31,C32,C33,C34,C35,C36]]
Entonces las matrices Ci se obtienen realizando los siguientes productos (los productos son renglón por renglón):

A below the text block is a message input field with placeholder "Pregúntame cualquier cosa" and a speech icon. To the right of the input field is a progress bar showing "0/4000" and a send button with a right-pointing arrow. On the far right of the window, there is a sidebar titled "Actividad reciente" listing recent interactions.

The screenshot shows a continuation of the Microsoft Bing Chat interface. The AI has listed several equations representing the calculation of matrix C row by row:

por renglón):
C1=A1xBT1
C2=A1xBT2
C3=A1xBT3
C4=A1xBT4
C5=A1xBT5
C6=A1xBT6
C7=A2xBT1
C8=A2xBT2
C9=A2xBT3
C10=A2xBT4
C11=A2xBT5
C12=A2xBT6
C13=A3xBT1
C14=A3xBT2
C15=A3xBT3
C16=A3xBT4
C17=A3xBT5
C18=A3xBT6
C19=A4xBT1
C20=A4xBT2
C21=A4xBT3

The rest of the window is identical to the first screenshot, showing the message input field, activity sidebar, and system status bar at the bottom.

The screenshot shows a Microsoft Bing AI Chat window. The user has asked the AI to analyze a task involving three nodes (Node 1, Node 2, and Node 3) performing matrix multiplication. The AI has provided detailed text-based responses for each node:

- Node 1:** Initializará las matrices A y B. Obtendrá la transpuesta de la matriz B. Enviará las matrices Ai y BTj al nodo 2 (solo enviará las matrices necesarias para realizar los productos C1 al C18). Enviará las matrices Ai y BTj al nodo 3 (solo enviará las matrices necesarias para realizar los productos C19 al C36). Recibirá del nodo 2 las matrices C1 a la C18. Recibirá del nodo 3 las matrices C19 a la C36. Obtendrá la matriz C a partir de las matrices C1 a la C36. Calculará y desplegará el checksum de la matriz C. Desplegará las matrices A, B y C para el caso N=6, M=5.
- Node 2:** Recibirá las matrices Ai y BTj del nodo 1 (solo recibirá las matrices necesarias para realizar los productos C1 al C18). Calculará los productos C1 al C18. Enviará las matrices C1 a la C18 al nodo 1.
- Node 3:** Recibirá las matrices Ai y BTj del nodo 1 (solo recibirá las matrices necesarias para realizar los productos C19 al C36).

The interface includes a sidebar for "Actividad reciente" (Recent activity) and a bottom status bar showing the date and time (12:47 p.m., 08/10/2023).

A lo que el chat nos responde con un análisis sobre lo que debe hacer cada nodo, pero todo reflejado en texto, sin código.

The screenshot shows the same Microsoft Bing AI Chat window. The AI has provided a general problem statement and specific steps for Node 1:

El problema que planteas es un ejemplo clásico de computación distribuida, donde se divide una tarea grande (en este caso, la multiplicación de matrices) en tareas más pequeñas que se pueden ejecutar en paralelo en diferentes nodos (máquinas virtuales en Azure en este caso). Aquí te dejo un esquema general de cómo podrías abordar este problema en Java:

1. Nodo 1 (Maestro):

- Inicializa las matrices A y B con los valores dados.
- Calcula la transpuesta de la matriz B.
- Divide las matrices A y BT en seis partes iguales utilizando la función `split` que proporcionaste.
- Envía las partes correspondientes de las matrices Ai y BTj a los nodos 2 y 3.
- Recibe las matrices C1 a C18 del nodo 2 y las matrices C19 a C36 del nodo 3.
- Combina todas las matrices Ci para formar la matriz C.
- Calcula y muestra el checksum de la matriz C.
- Muestra las matrices A, B y C para el caso N=6, M=5.

The interface includes a sidebar for "Actividad reciente" (Recent activity) and a bottom status bar showing the date and time (12:48 p.m., 08/10/2023).

Chatear con GPT-4 en Bing

https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1&nref=1

Microsoft Bing BÚSQUEDA CHAT

English Javatt 2042 Actividad reciente

Combinas todas las matrices Ci para formar la matriz C.

Calcula y muestra el checksum de la matriz C.

Muestra las matrices A, B y C para el caso N=6, M=5.

2. Nodo 2 (Trabajador):

- Recibe las partes correspondientes de las matrices Ai y BTj del nodo 1.
- Calcula los productos C1 a C18.
- Envía las matrices C1 a C18 al nodo 1.

3. Nodo 3 (Trabajador):

- Recibe las partes correspondientes de las matrices Ai y BTj del nodo 1.
- Calcula los productos C19 a C36.
- Envía las matrices C19 a C36 al nodo 1.

Para implementar esto en Java, necesitarás utilizar alguna forma de comunicación entre procesos, como sockets, para enviar y recibir datos entre los nodos. También necesitarás manejar la concurrencia y la sincronización para asegurarte de que todas las operaciones se realicen en el orden correcto. Por último, necesitarás implementar la lógica para dividir las matrices, calcular los productos y combinar los resultados.

Espero que esto te ayude a empezar con tu tarea. Si tienes más preguntas, no dudes en preguntar.

Nuevo tema Pregúntame cualquier cosa 0/4000 Comentarios

12:48 p.m. 08/10/2023

Por lo que le pedimos que nos proporcione el código del problema que le hemos planteado.

Chatear con GPT-4 en Bing

https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1&nref=1

Microsoft Bing BÚSQUEDA CHAT

English Javatt 2042 Actividad reciente

Genera el código en java para el problema que planteé

Aquí tienes un esquema básico de cómo podrías implementar esto en Java. Ten en cuenta que este código es solo un esquema y no incluye detalles como la inicialización de las matrices A y B, la comunicación entre procesos o el manejo de errores.

Java

Este código se genera por IA. Revise y use cuidadosamente. Visite nuestras preguntas más frecuentes para obtener más información.

```
// Nodo 1 (Maestro)
public class Nodo1 {
    public static void main(String[] args) {
        // Inicializar las matrices A y B
        double[][] A = ...;
        double[][] B = ...;

        // Calcular la transpuesta de B
        double[][] BT = transpose(B);

        // Dividir las matrices A y BT en seis partes iguales
    }
}
```

Nuevo tema Pregúntame cualquier cosa 0/4000 Comentarios

12:48 p.m. 08/10/2023

Chatear con GPT-4 en Bing

https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1&nref=1

Microsoft Bing BÚSQUEDA CHAT

English Javatt 2042 Actividad reciente

```

// Dividir las matrices A y BT en seis partes iguales
double[][][] A_split = split(A);
double[][][] BT_split = split(BT);

// Enviar las partes correspondientes de las matrices Ai y BTj a los nodos 2 y 3
// ...

// Recibir las matrices C1 a C18 del nodo 2 y las matrices C19 a C36 del nodo 3
// ...

// Combinar todas las matrices Ci para formar la matriz C
double[][] C = combine(C_split);

// Calcular y mostrar el checksum de la matriz C
double checksum = calculateChecksum(C);
System.out.println("Checksum: " + checksum);

// Mostrar las matrices A, B y C para el caso N=6, M=5
// ...
}
}

```

Nuevo tema Pregúntame cualquier cosa 0/4000 Comentarios

Te proporcionare un codigo por partes y 5:38 p.m.

Tarea 3 1:28 a.m.

Producto de matrices MxN 0:11 a.m.

Desde ahora trabajaremos con un proy... Hace 4 días

12:49 p.m. 08/10/2023

Como vemos el programa que nos proporciona es parcial, nos genera la estructura de los nodos 1, 2 y 3. Por lo que le pediremos que nos genere las funciones para cada requerimiento que queramos hacer como dividir las matrices, juntarlas, imprimirlas en pantalla, etc.

Chatear con GPT-4 en Bing

https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1&nref=1

Microsoft Bing BÚSQUEDA CHAT

English Javatt 2042 Actividad reciente

```

// Nodo 2 (Trabajador)
public class Nodo2 {
    public static void main(String[] args) {
        // Recibir las partes correspondientes de las matrices Ai y BTj del nodo 1
        // ...

        // Calcular los productos C1 a C18
        double[][][] C_split = calculateProducts(A_split, BT_split);

        // Enviar las matrices C1 a C18 al nodo 1
        // ...
    }
}

// Nodo 3 (Trabajador)
public class Nodo3 {
    public static void main(String[] args) {
        // Recibir las partes correspondientes de las matrices Ai y BTj del nodo 1
        // ...

        // Calcular los productos C19 a C36
        double[][][] C_split = calculateProducts(A_split, BT_split);
    }
}

```

Nuevo tema Pregúntame cualquier cosa 0/4000 Comentarios

Te proporcionare un codigo por partes y 5:38 p.m.

Tarea 3 1:28 a.m.

Producto de matrices MxN 0:11 a.m.

Desde ahora trabajaremos con un proy... Hace 4 días

12:49 p.m. 08/10/2023

Función para generar la transpuesta de la matriz B.

The screenshot shows the Microsoft Bing Chat interface. The user has asked for Java code to generate the transpose of a matrix B. The AI has provided the following code:

```
public static double[][] transpose(double[][] B) {
    int n = B.length; // Número de filas de la matriz B
    int m = B[0].length; // Número de columnas de la matriz B

    // Crear una nueva matriz para almacenar la transpuesta
    double[][] BT = new double[m][n];

    // Transponer la matriz
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            BT[j][i] = B[i][j];
        }
    }

    return BT;
}
```

The interface includes a sidebar with activity recent items like "Tarea 3" and "Producto de matrices MxN". A bottom bar shows system icons and the date/time (12:51 p.m., 08/10/2023).

Función que divide en 6 partes iguales una matriz.

The screenshot shows the Microsoft Bing Chat interface. The user has asked for Java code to divide a rectangular matrix A into six equal parts. The AI has provided the following code:

```
public static double[][][] split(double[][] A) {
    int n = A.length; // Número de filas de la matriz A
    int m = A[0].length; // Número de columnas de la matriz A

    // Calculamos el tamaño de cada parte de la matriz dividida
    int partSize = n / 6;

    // Dividimos la matriz A en seis partes de igual tamaño
    double[][][] A_split = new double[6][][];
    for (int i = 0; i < 6; i++) {
        A_split[i] = new double[partSize][m];
    }

    // Copiamos los datos de la matriz A a las partes divididas
    int offset = 0;
    for (int i = 0; i < 6; i++) {

```

The interface includes a sidebar with activity recent items like "Tarea 3" and "Producto de matrices MxN". A bottom bar shows system icons and the date/time (12:53 p.m., 08/10/2023).

Chatear con GPT-4 en Bing

Microsoft Bing BÚSQUEDA CHAT

```

// Calculamos el tamaño de cada parte de la matriz dividida
int partSize = n / 6;

// Dividimos la matriz A en seis partes de igual tamaño
double[][][] A_split = new double[6][][];
for (int i = 0; i < 6; i++) {
    A_split[i] = new double[partSize][m];
}

// Copiamos los datos de la matriz A a las partes divididas
int offset = 0;
for (int i = 0; i < 6; i++) {
    for (int j = 0; j < partSize; j++) {
        System.arraycopy(A[offset], 0, A_split[i][j], 0, m);
        offset++;
    }
}

return A_split;
}

```

Esta función toma como entrada una matriz `A` y devuelve una matriz tridimensional `A_split` que contiene las seis partes en las que se dividió `A`. Cada parte tiene el mismo número de filas (`partSize`) y el mismo número de columnas (`m`) que la matriz original. La función `System.arraycopy` se utiliza para copiar los datos de la matriz `A` a las partes divididas. Espero que esto te ayude! Si tienes más preguntas, no dudes en preguntar.

Nuevo tema Pregúntame cualquier cosa 0/4000 Comentarios

English Javatt 2042 Actividad reciente

Te proporcionare un código por partes y 5:38 p.m.
Tarea 3 1:28 a.m.
Producto de matrices MxN 0:11 a.m.
Desde ahora trabajaremos con un pro... Hace 4 días

Búsqueda Búsquedas

12:53 p.m. 08/10/2023

Función que inicializa las matrices A y B. En esta función, al crear la matriz B tiene un error en los índices ya que A es MxN y B NxM, así que lo tendremos que arreglar después.

Chatear con GPT-4 en Bing

Microsoft Bing BÚSQUEDA CHAT

Genera un método que inicialice las matrices de la siguiente manera:
 $A[i][j] = 5*i - 2*j$
 $B[i][j] = 6*i + 3*j$

Aquí tienes dos métodos en Java que puedes usar para inicializar las matrices A y B de acuerdo a las fórmulas que proporcionaste:

Java
Este código se genera por IA. Revise y use cuidadosamente. Visite nuestras preguntas más frecuentes para obtener más información.

```

public static double[][] initializeA(int n, int m) {
    double[][] A = new double[n][m];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            A[i][j] = 5 * i - 2 * j;
        }
    }
    return A;
}

public static double[][] initializeB(int n, int m) {
    double[][] B = new double[n][m];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            B[i][j] = 6 * i + 3 * j;
        }
    }
    return B;
}

```

Nuevo tema Pregúntame cualquier cosa 0/4000 Comentarios

English Javatt 2042 Actividad reciente

Te proporcionare un código por partes y 5:38 p.m.
Tarea 3 1:28 a.m.
Producto de matrices MxN 0:11 a.m.
Desde ahora trabajaremos con un pro... Hace 4 días

Búsqueda Búsquedas

12:55 p.m. 08/10/2023

Función para generar el checksum de una matriz.

The screenshot shows the Microsoft Bing AI Chat interface. The user has asked for a function to calculate the checksum of a matrix. The AI has provided the following Java code:

```
Java
Este código se genera por IA. Revise y use cuidadosamente. Visite nuestras preguntas más frecuentes para obtener más información.

public static double calculateChecksum(double[][] C) {
    double checksum = 0.0;
    for (int i = 0; i < C.length; i++) {
        for (int j = 0; j < C[0].length; j++) {
            checksum += C[i][j];
        }
    }
    return checksum;
}
```

A note below the code states: "Esta función toma como entrada una matriz C y devuelve su checksum, que se calcula como la suma de todos los elementos de la matriz. Espero que esto te ayude! Si tienes más preguntas, no dudes en preguntar."

The interface also shows a message history on the right side:

- Te proporcionare un código por partes y 5:38 p.m.
- Tarea 3 1:28 a.m.
- Producto de matrices MxN 0:11 a.m.
- Desde ahora trabajaremos con un proy... Hace 4 días

Función para juntar las matrices.

The screenshot shows the Microsoft Bing AI Chat interface. The user has asked for a function to combine matrix parts. The AI has provided the following Java code:

```
Java
Este código se genera por IA. Revise y use cuidadosamente. Visite nuestras preguntas más frecuentes para obtener más información.

public static double[][] combine(double[][][] C_split) {
    int partSize = C_split[0].length; // Número de filas de cada parte
    int m = C_split[0][0].length; // Número de columnas de cada parte

    // Crear una nueva matriz para almacenar la matriz combinada
    double[][] C = new double[partSize][m];

    // Combinar las partes de la matriz C en la matriz C
    int offset = 0;
    for (int i = 0; i < partSize; i++) {
        for (int j = 0; j < partSize; j++) {
            System.arraycopy(C_split[i][j], 0, C[offset], 0, m);
            offset++;
        }
    }
}
```

The interface also shows a message history on the right side:

- Te proporcionare un código por partes y 5:38 p.m.
- Tarea 3 1:28 a.m.
- Producto de matrices MxN 0:11 a.m.
- Desde ahora trabajaremos con un proy... Hace 4 días

Función para multiplicar matrices renglón por renglón, en esta petición fue necesario ser muy específico en como se distribuyen las multiplicaciones en C, ya que, si se le solicita de forma normal, casi siempre genera una multiplicación de renglón por columna.

Chatear con GPT-4 en Bing

https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1&ntref=1

Microsoft Bing BÚSQUEDA CHAT

English Javatt 2042 Actividad reciente

Crea una función en java que realice la multiplicación de matrices de la siguiente manera:
 -La matrices que se multiplicaran serán del tipo double y las operaciones deben hacerse con arreglos en java.
 -La validación para hacer esta multiplicación de matrices es que ambas matrices tengan el mismo número de columnas.
 -Sea A=[A1 A2 A3 A4, A5 A6 A7 A8] una matriz de n=2 (filas) y m=4 (columnas) y Bt(transpuesta de B)= [B1,B2,B3,B4, [B5,B6,B7,B8], [B9,B10,B11,B12]] una matriz de k=3(filas) y m=4(columnas).
 Entonces el resultado de multiplicar estas dos es C que es una matriz de n*m.
 -C es igual a: [(A1*B1+A2*B2+A3*B3+A4*B4),(A1*B5+A2*B6+A3*B7+A4*B8),
 (A1*B9+A2*B10+A3*B11+A4*B12)],
 [(A5*B1+A6*B2+A7*B3+A8*B4),(A5*B5+A6*B6+A7*B7+A8*B8),(A5*B9+A6*B10+A7*B11+A8*B12)]]

Aquí tienes una función en Java que puedes usar para multiplicar dos matrices de la manera que describiste:

```
Java
Este código se genera por IA. Revise y use cuidadosamente. Visite nuestras preguntas más frecuentes para obtener más información.

public static double[][] multiply(double[][] A, double[][] BT) {
    int n = A.length; // Número de filas de la matriz A
    int m = A[0].length; // Número de columnas de la matriz A
    int k = BT.length; // Número de filas de la matriz BT

    // Verificar si las matrices tienen el mismo número de columnas
}
```

Nuevo tema Pregúntame cualquier cosa 0/4000 Comentarios

01:00 p.m. 08/10/2023

Chatear con GPT-4 en Bing

https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1&ntref=1

Microsoft Bing BÚSQUEDA CHAT

English Javatt 2042 Actividad reciente

```
if (m != BT[0].length) {
    throw new IllegalArgumentException("Las matrices deben tener el mismo número de columnas");
}

// Crear una nueva matriz para almacenar el producto
double[][] C = new double[n][k];

// Multiplicar las matrices renglón por renglón
for (int i = 0; i < n; i++) {
    for (int j = 0; j < k; j++) {
        for (int l = 0; l < m; l++) {
            C[i][j] += A[i][l] * BT[j][l];
        }
    }
}

return C;
}
```

Esta función toma como entrada dos matrices `A` y `BT`, y devuelve su producto `C`. La multiplicación de matrices se realiza renglón por renglón, es decir, cada elemento `C[i][j]` se calcula como la suma de los productos `A[i][l] * BT[j][l]` para todos los `l`. Si las matrices no tienen el mismo número de columnas, la función lanza una excepción. Espero que esto te ayude! Si tienes más preguntas, no dudes en preguntar.

Nuevo tema Pregúntame cualquier cosa 0/4000 Comentarios

01:03 p.m. 08/10/2023

Por último, generamos una función que imprime las matrices en pantalla.

A screenshot of a Microsoft Bing search results page. The URL in the address bar is <https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1&tref=1>. The search query was "Por ultimo generame una funcion que me me imprime en pantalla un arreglo de dos dimensiones.". The results show a generated Java code snippet:

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

The code prints a 2D matrix row by row. Below the code, there is a note explaining its functionality:

Esta función toma como entrada un arreglo bidimensional `matrix` y lo imprime en pantalla. Recorre cada fila de la matriz y, dentro de cada fila, recorre cada columna, imprimiendo el valor de cada elemento seguido de un espacio. Al final de cada fila, imprime una línea nueva para separar las filas. Espero que esto te ayude! Si tienes más preguntas, no dudes en preguntar.

The Bing interface shows other recent activity and a comment section at the bottom right.

Pasamos a generar el código del nodo 1,2,3 y el main.

A screenshot of a Microsoft Bing search results page. The URL in the address bar is <https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1&tref=1>. The search query was "en este programa el usuario debe pasar el numero de nodo que se va a ejecutar por consola, si es 1 se ejecuta el codigo del nodo 1, si es 2 o 3 se ejecuta el código el código de los nodos 2 y 3.". The results show a generated Java Main class:

```
public class Main {
    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Por favor, pasa el número de nodo como argumento.");
            return;
        }

        int nodo;
        try {
            nodo = Integer.parseInt(args[0]);
        } catch (NumberFormatException e) {
            System.out.println("El argumento debe ser un número entero.");
            return;
        }
    }
}
```

The code checks if a single argument is passed on the command line. If it is, it prints a message and exits. Otherwise, it tries to parse the first argument as an integer and sets it as the `nodo` variable. A note above the code explains the requirement:

en este programa el usuario debe pasar el numero de nodo que se va a ejecutar por consola, si es 1 se ejecuta el código del nodo 1, si es 2 o 3 se ejecuta el código el código de los nodos 2 y 3.

The Bing interface shows other recent activity and a comment section at the bottom right.

Este código valida que el usuario haya puesto un argumento al ejecutar el programa y este sea un número, después, despendiendo de que numero puso, ejecuta el código del nodo 1 o en su defecto el nodo2-3.

The screenshot shows a Microsoft Bing search results page with the URL <https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1&ntref=1>. The search bar contains "Chatear con GPT-4 en Bing". The page displays a Java code snippet:switch (nodo) { case 1: // Ejecutar el código del Nodo 1 Nodo1.main(args); break; case 2: // Ejecutar el código del Nodo 2 Nodo2.main(args); break; case 3: // Ejecutar el código del Nodo 3 Nodo3.main(args); break; default: System.out.println("Número de nodo no válido. Debe ser 1, 2 o 3."); break;}A note below the code states: "Este código toma el primer argumento de línea de comando, lo convierte en un número entero y luego ejecuta el código correspondiente al número de nodo. Si el argumento no es un número entero o si no es 1, 2 o 3, se muestra un mensaje de error." Another note says: "Para ejecutar este programa con un argumento de línea de comando, puedes usar el siguiente comando en la terminal:". A chat interface at the bottom allows users to ask questions and receive answers from AI.

De esta consulta solo obtendremos la estructura básica, debido que ocurrieron un par de errores al momento de usarlo que arreglamos después.

The screenshot shows a Microsoft Bing search results page with the same URL as the previous one. A callout box from the AI interface provides instructions: "Al código anterior, agrega la lógica para que el nodo 1 reciba la multiplicación de las matrices que le proporcionaste al nodo 2 y 3, una vez recuperadas todas las partes, juntarlas en una sola matriz (C) y calcular su checksum." Below this, a note says: "Aquí está el código del Nodo 1 (Maestro) actualizado para recibir las matrices resultantes de los nodos 2 y 3, combinarlas en una sola matriz y calcular su checksum." The code block now includes matrix multiplication logic:import java.io.*;
import java.net.*;

public class Nodo1 {
 public static void main(String[] args) {
 // Inicializar las matrices A y B
 double[][] A = ...;
 double[][] B = ...;

 // Calcular la transpuesta de B
 double[][] BT = transpose(B);

 // Dividir las matrices A y BT en seis partes iguales
 }
}The rest of the page remains the same, showing the original Java code for validating a command-line argument and executing different nodes based on its value.

Chatear con GPT-4 en Bing

https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1&ntref=1

Microsoft Bing BÚSQUEDA CHAT

English Javatt 2042 Actividad reciente

```
// Dividir las matrices A y BT en seis partes iguales
double[][][] A_split = split(A);
double[][][] BT_split = split(BT);

try {
    // Crear sockets para los nodos 2 y 3.
    Socket socket2 = new Socket("localhost", 5000); // Reemplaza "localhost" con la
    dirección del Nodo 2
    Socket socket3 = new Socket("localhost", 5001); // Reemplaza "localhost" con la
    dirección del Nodo 3

    // Crear flujos de salida para los nodos 2 y 3
    ObjectOutputStream out2 = new ObjectOutputStream(socket2.getOutputStream());
    ObjectOutputStream out3 = new ObjectOutputStream(socket3.getOutputStream());

    // Enviar las primeras 3 partes de las matrices Ai y BTj al nodo 2
    for (int i = 0; i < 3; i++) {
        out2.writeObject(A_split[i]);
        out2.writeObject(BT_split[i]);
    }

    // Enviar las últimas 3 partes de las matrices Ai y BTj al nodo 3
    for (int i = 3; i < 6; i++) {
        out3.writeObject(A_split[i]);
        out3.writeObject(BT_split[i]);
    }
}
```

Nuevo tema Pregúntame cualquier cosa 0/4000 Comentarios

01:17 p.m. 08/10/2023

Chatear con GPT-4 en Bing

https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1&ntref=1

Microsoft Bing BÚSQUEDA CHAT

English Javatt 2042 Actividad reciente

```
// Crear flujos de entrada para los nodos 2 y 3
ObjectInputStream in2 = new ObjectInputStream(socket2.getInputStream());
ObjectInputStream in3 = new ObjectInputStream(socket3.getInputStream());

// Recibir las matrices resultantes de los nodos 2 y 3
double[][][] C_split = new double[6][][];
for (int i = 0; i < 3; i++) {
    C_split[i] = (double[][][]) in2.readObject();
}
for (int i = 3; i < 6; i++) {
    C_split[i] = (double[][][]) in3.readObject();
}

// Combinar todas las matrices Ci para formar la matriz C
double[][] C = combine(C_split);

// Calcular y mostrar el checksum de la matriz C
double checksum = calculateChecksum(C);
System.out.println("Checksum: " + checksum);

// Cerrar los flujos de entrada y salida y los sockets
in2.close();
in3.close();
out2.close();
out3.close();
```

Nuevo tema Pregúntame cualquier cosa 0/4000 Comentarios

01:18 p.m. 08/10/2023

Estructura básica para el nodo 2 y 3. Nos genera el código de un servidor TCP multithread.

A screenshot of a Microsoft Bing search results page. The search query is "Estructura básica para el nodo 2 y 3. Nos genera el código de un servidor TCP multithread.". The results page shows a snippet of Java code for a TCP server node. The code implements a basic structure for handling multiple clients. It includes creating a server socket, accepting connections, and starting a new thread for each client. The code also handles exceptions and reads/writes data using ObjectInputStream and ObjectOutputStream. The Bing interface includes a chat window at the bottom where the user can ask follow-up questions.

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

public class Nodo {
    public static void main(String[] args) {
        try {
            // Crear un socket de servidor para escuchar las conexiones entrantes
            ServerSocket serverSocket = new ServerSocket(5000); // Reemplaza 5000 con el puerto del nodo

            while (true) {
                // Aceptar una conexión entrante
                Socket socket = serverSocket.accept();

```

A screenshot of a Microsoft Bing search results page, continuing from the previous one. It shows the completion of the Java code for the TCP server node. The code now includes the Worker class, which extends Thread and handles the actual multiplication of matrices. It uses ObjectInputStream and ObjectOutputStream to read matrices from the client and write the result back. The code demonstrates how to calculate the product of two matrices (A1 x B1) and send the result back to the client. The Bing interface remains the same, with a chat window at the bottom.

```
        // Crear un nuevo hilo para manejar la conexión
        new Worker(socket).start();
    }
} catch (IOException e) {
    e.printStackTrace();
}

private static class Worker extends Thread {
    private Socket socket;

    public Worker(Socket socket) {
        this.socket = socket;
    }

    public void run() {
        try {
            // Crear un flujo de entrada para recibir las matrices A1 y B1
            ObjectInputStream in = new ObjectInputStream(socket.getInputStream());

            // Recibir las matrices A1 y B1
            double[][] A1 = (double[][]) in.readObject();
            double[][] B1 = (double[][]) in.readObject();

            // Calcular el producto C1 = A1 x B1
            double[][] C1 = multiply(A1, B1);

            // Crear un flujo de salida para enviar la matriz C1
            ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());

```

El problema que se presentó fue que el cliente no mandaba bien las matrices al servidor y se quedaba pasmado, por eso consultaremos al chat para que nos demuestre genere de nuevo este fragmento del programa. Para esto le pasamos la estructura del código.

Chatear con GPT-4 en Bing

https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1&ntref=1

Microsoft Bing BÚSQUEDA CHAT

English Javatt 2042 Actividad reciente

```
parte 3: cliente (C++ y servidor TCP // 10.0.0.1:5000). Función que ejecuta el nodo 1
private static void executeNode1() throws Exception {
    // Inicializar las matrices A y B
    double[][] A = initializeMatrix(N, M, true);
    double[][] B = initializeMatrix(N, M, false);

    // Calcular la transpuesta de B
    double[][] BT = transpose(B);

    // Dividir las matrices A y BT en seis partes iguales
    double[][][] A_split = split(A);

    try {
        Socket socket2 = new Socket("20.75.158.31", 50000);
        Socket socket3 = new Socket("", 50000);

        socket2.close();
        socket3.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

// NOTE: Función que ejecuta el nodo 2 y 3
private static void executeNode2or3(int nodeNumber) throws Exception {

    ServerSocket serverSocket = new ServerSocket(50000); // Reemplaza 5000 con el puerto del nodo
    System.out.println("Servidor iniciado en el puerto: " + 50000);
    while (true) {
        Socket socket = serverSocket.accept();
    }
}
```

Nuevo tema Pregúntame cualquier cosa 0/4000 Comentarios 01:20 p.m. 08/10/2023

Chatear con GPT-4 en Bing

https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1&ntref=1

Microsoft Bing BÚSQUEDA CHAT

English Javatt 2042 Actividad reciente

```
System.out.println("Servidor iniciado en el puerto: " + 50000);
while (true) {
    Socket socket = serverSocket.accept();
    new Worker(socket).start();
}

private static class Worker extends Thread {
    private Socket socket;

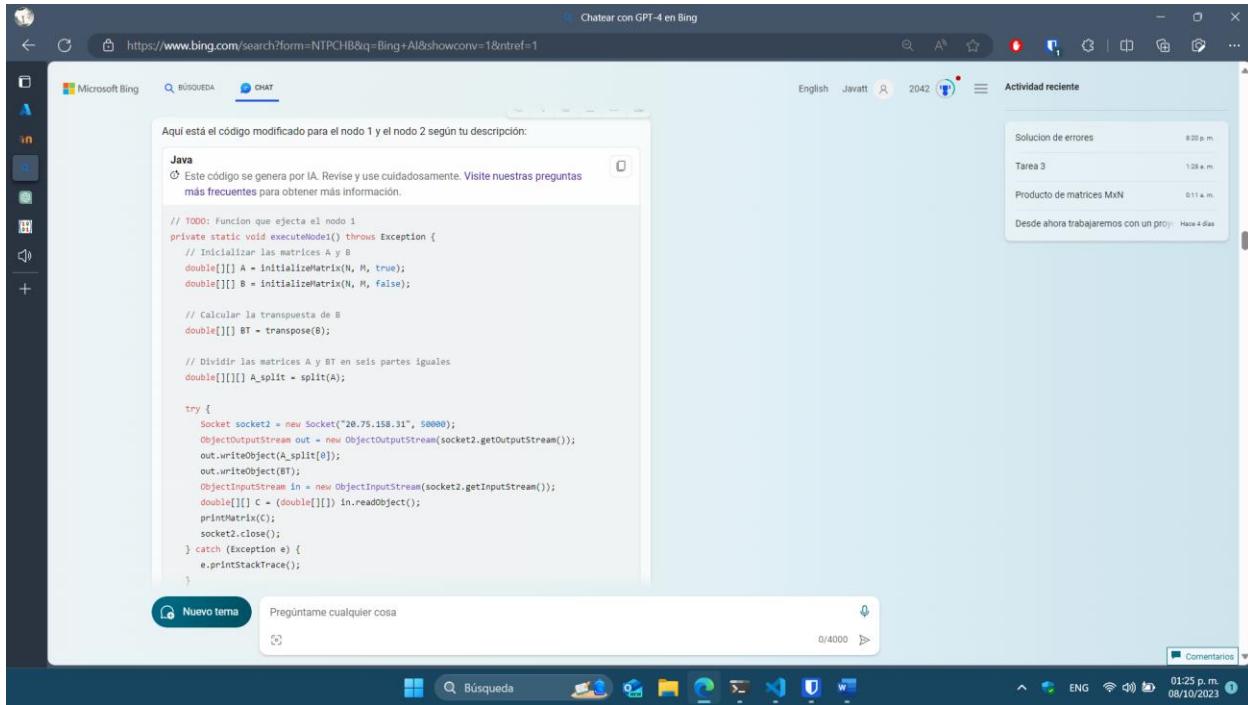
    public Worker(Socket socket) {
        this.socket = socket;
    }

    public void run() {
        try {
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

En esta última parte, quiero que generes en el código del nodo 1 la primera matriz de `A_split[0]` y toda la matriz `bt` hacia el nodo 2, el nodo 2, en el `run()` deberá obtener estas matrices y una vez recibidas, deberá multiplicarlas con `multiply`, una vez multiplicadas deberá enviar al nodo 1 el resultado de esta multiplicación, el nodo 1 deberá obtener el resultado e imprimirla en pantalla.

Nuevo tema Pregúntame cualquier cosa 0/4000 Comentarios 01:25 p.m. 08/10/2023

Este código del nodo 1 efectivamente manda un fragmento de la matriz A y la transpuesta de la matriz B al nodo 2.



Aqui está el código modificado para el nodo 1 y el nodo 2 según tu descripción:

```
Java
Este código se genera por IA. Revise y use cuidadosamente. Visite nuestras preguntas más frecuentes para obtener más información.

// TODO: Función que ejecuta el nodo 1
private static void executeNode1() throws Exception {
    // Inicializar las matrices A y B
    double[][] A = initializeMatrix(N, M, true);
    double[][] B = initializeMatrix(N, M, false);

    // Calcular la transpuesta de B
    double[][] BT = transpose(B);

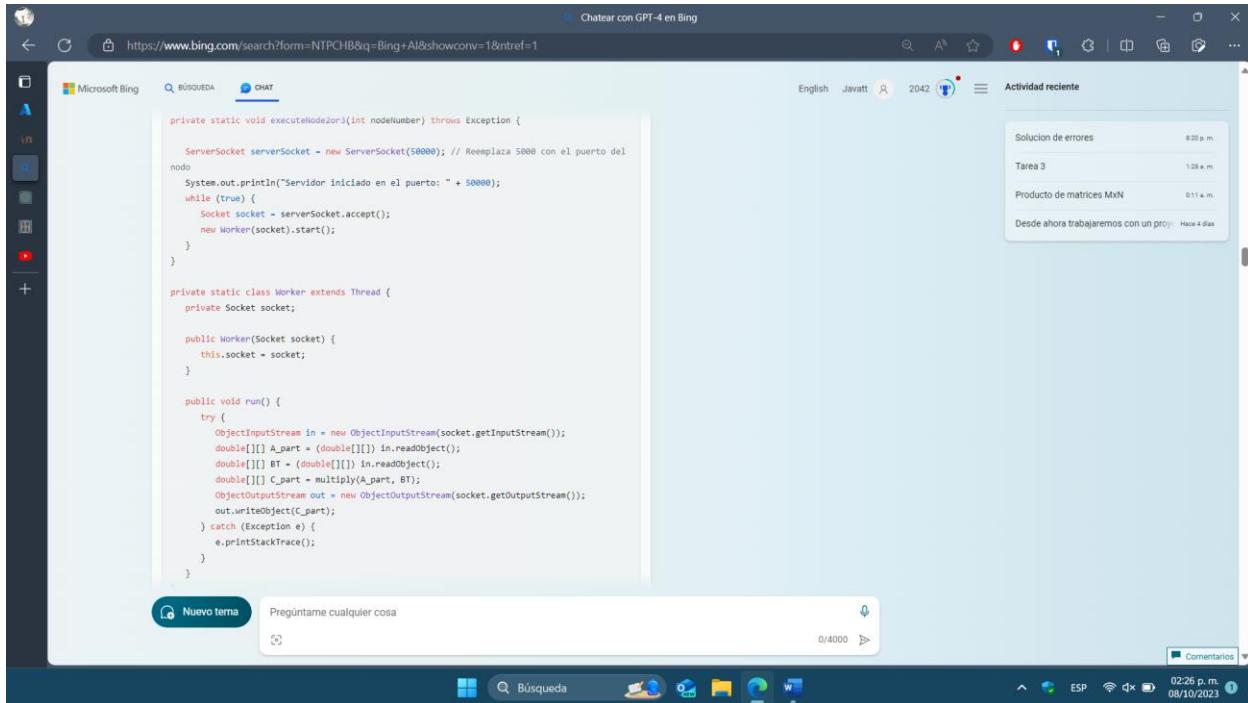
    // Dividir las matrices A y BT en seis partes iguales
    double[][][] A_split = split(A);

    try {
        Socket socket2 = new Socket("20.75.158.31", 50000);
        ObjectOutputStream out = new ObjectOutputStream(socket2.getOutputStream());
        out.writeObject(A_split[0]);
        out.writeObject(BT);
        ObjectInputStream in = new ObjectInputStream(socket2.getInputStream());
        double[][] C = (double[][]) in.readObject();
        printMatrix(C);
        socket2.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Nuevo tema Pregúntame cualquier cosa 0/4000

01:25 p. m. 08/10/2023

En el código del nodo 2 se reciben estas dos matrices, se hace la multiplicación y se manda al nodo 1 nuevamente. Ahora solo falta indicarle que a cada nodo se le enviaran ciertos fragmentos de A para que haga el mismo proceso.



```
Java
Este código se genera por IA. Revise y use cuidadosamente. Visite nuestras preguntas más frecuentes para obtener más información.

private static void executeNode2(int nodeNumber) throws Exception {

    ServerSocket serverSocket = new ServerSocket(50000); // Reemplaza 5000 con el puerto del nodo
    System.out.println("Servidor iniciado en el puerto: " + 50000);
    while (true) {
        Socket socket = serverSocket.accept();
        new Worker(socket).start();
    }
}

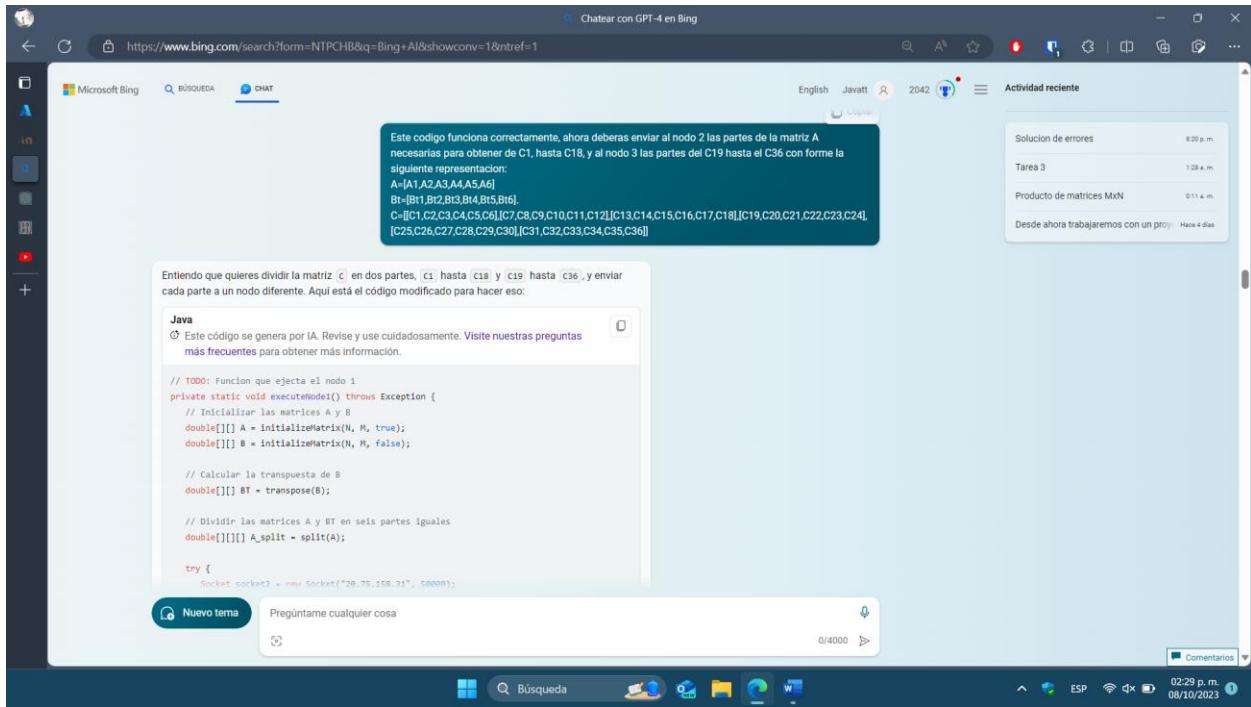
private static class Worker extends Thread {
    private Socket socket;
}

public Worker(Socket socket) {
    this.socket = socket;
}

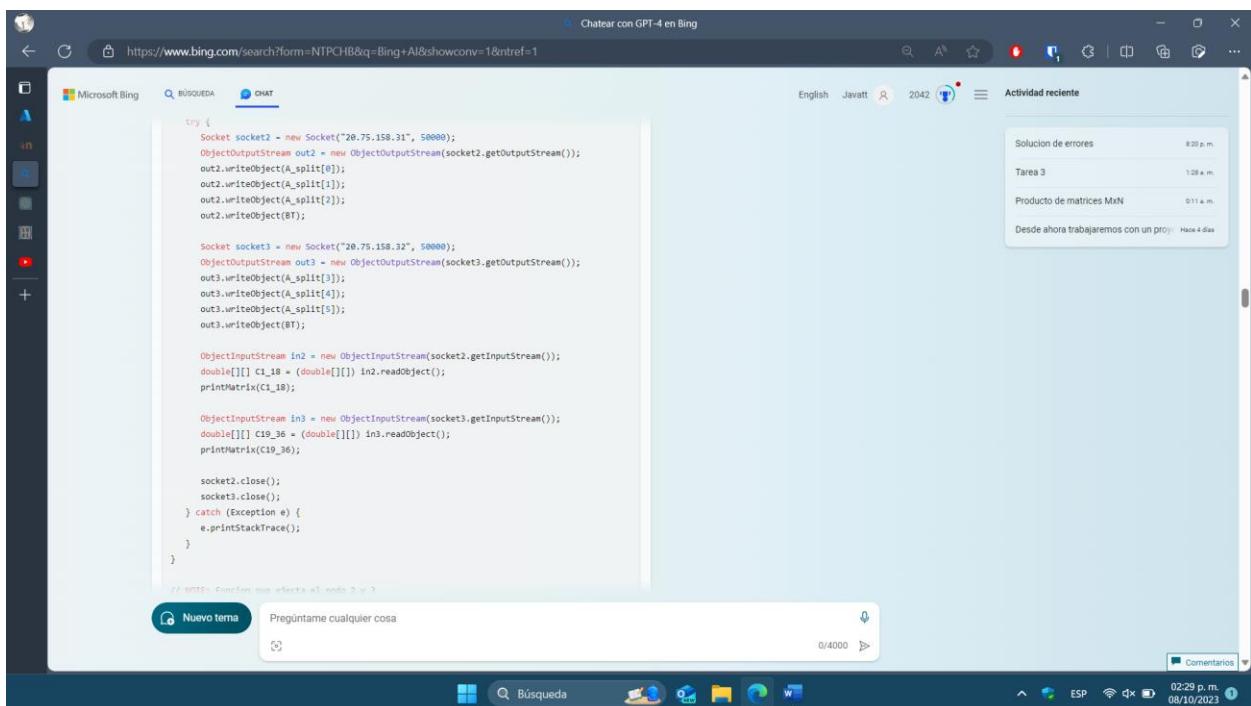
public void run() {
    try {
        ObjectInputStream in = new ObjectInputStream(socket.getInputStream());
        double[][] A_part = (double[][]) in.readObject();
        double[][] BT = (double[][]) in.readObject();
        double[][] C_part = multiply(A_part, BT);
        ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());
        out.writeObject(C_part);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Nuevo tema Pregúntame cualquier cosa 0/4000

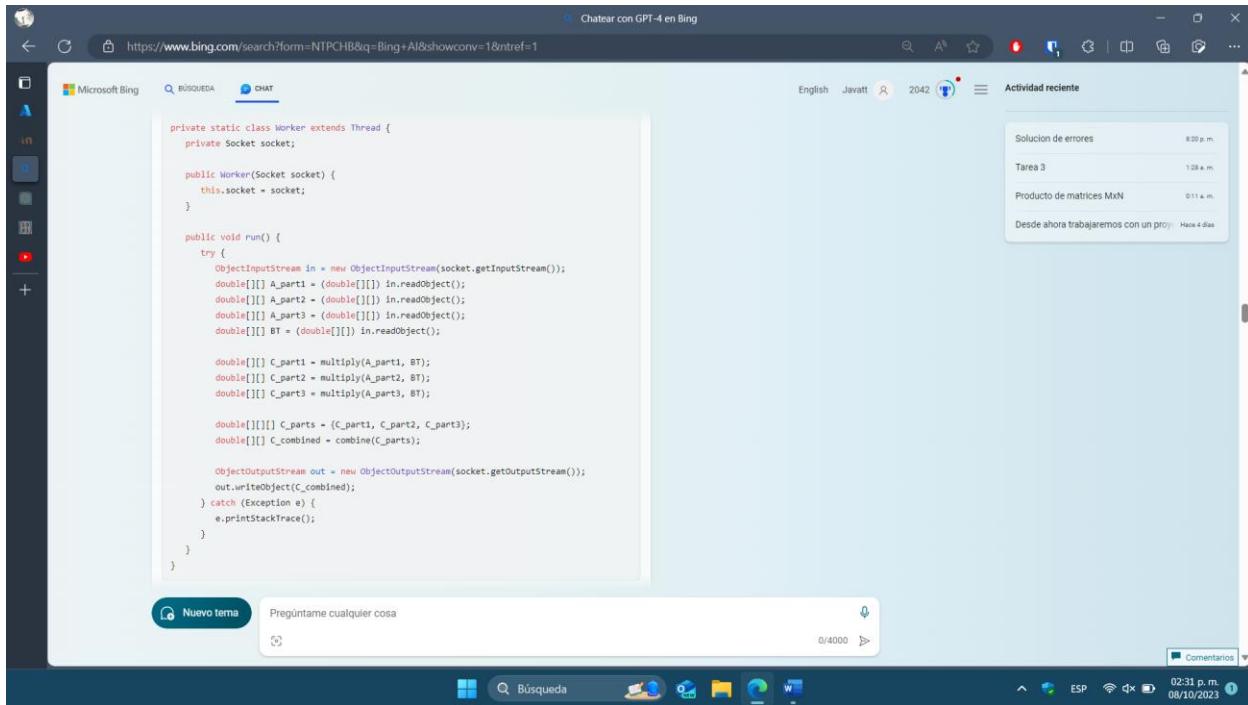
02:26 p. m. 08/10/2023



Podemos ver que ahora el cliente manda los respectivos fragmentos a cada servidor.



El problema que surgió con este código fue que, al servidor intentar juntar las partes que calculo y enviarlas, el cliente no las recibía, por lo que optaremos por enviar cada resultado por separado.



The screenshot shows the Microsoft Bing Chat interface. The search bar at the top contains the URL <https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1&nref=1>. The main area displays a Java code snippet for matrix multiplication:

```
private static class Worker extends Thread {
    private Socket socket;

    public Worker(Socket socket) {
        this.socket = socket;
    }

    public void run() {
        try {
            ObjectInputStream in = new ObjectInputStream(socket.getInputStream());
            double[][] A_part1 = (double[][]) in.readObject();
            double[][] A_part2 = (double[][]) in.readObject();
            double[][] A_part3 = (double[][]) in.readObject();
            double[][] BT = (double[][]) in.readObject();

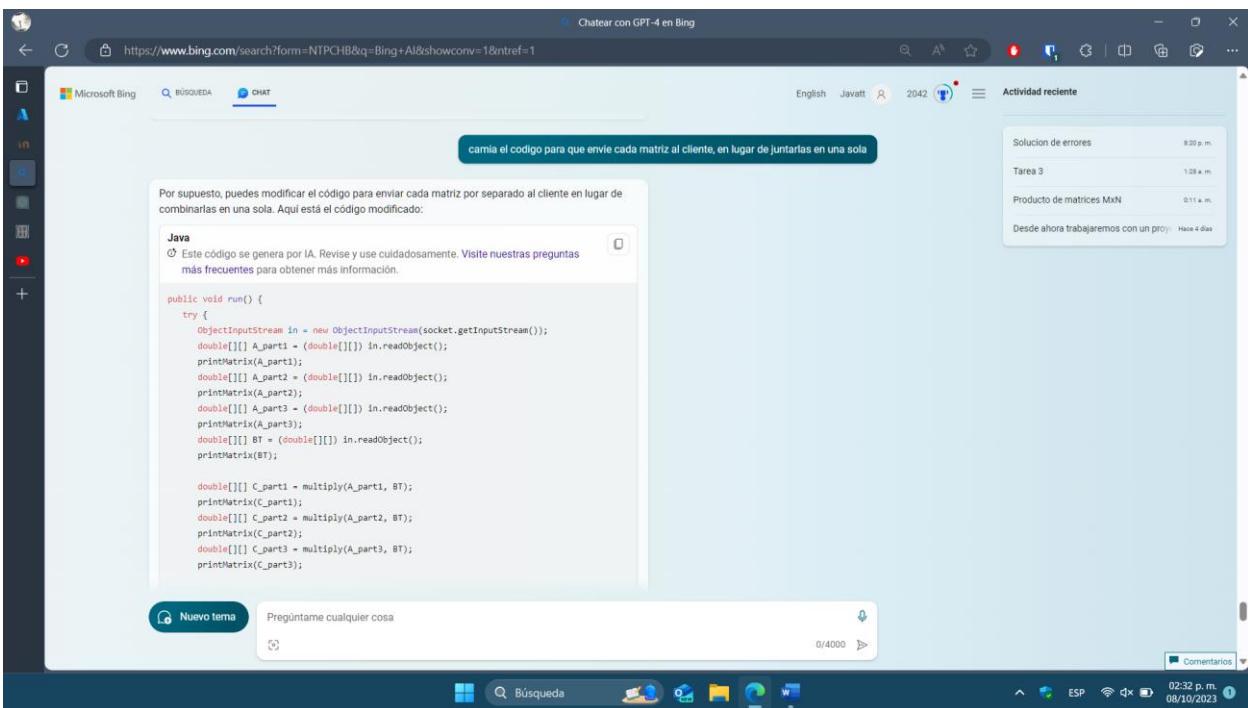
            double[][] C_part1 = multiply(A_part1, BT);
            double[][] C_part2 = multiply(A_part2, BT);
            double[][] C_part3 = multiply(A_part3, BT);

            double[][][] C_parts = {C_part1, C_part2, C_part3};
            double[][] C_combined = combine(C_parts);

            ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());
            out.writeObject(C_combined);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Below the code, there is a message input field with the placeholder "Pregúntame cualquier cosa" and a character limit of "0/4000". The bottom right corner of the window shows the date and time as "02:31 p. m. 08/10/2023".

Ahora en el servidor se envía cada parte de la matriz calculada al cliente de nuevo.



The screenshot shows the Microsoft Bing Chat interface again. The search bar at the top contains the same URL. A blue bar at the top of the main content area says "cambia el código para que envíe cada matriz al cliente, en lugar de juntarlas en una sola". Below this, a message from GPT-4 says: "Por supuesto, puedes modificar el código para enviar cada matriz por separado al cliente en lugar de combinarlas en una sola. Aquí está el código modificado:"

Java

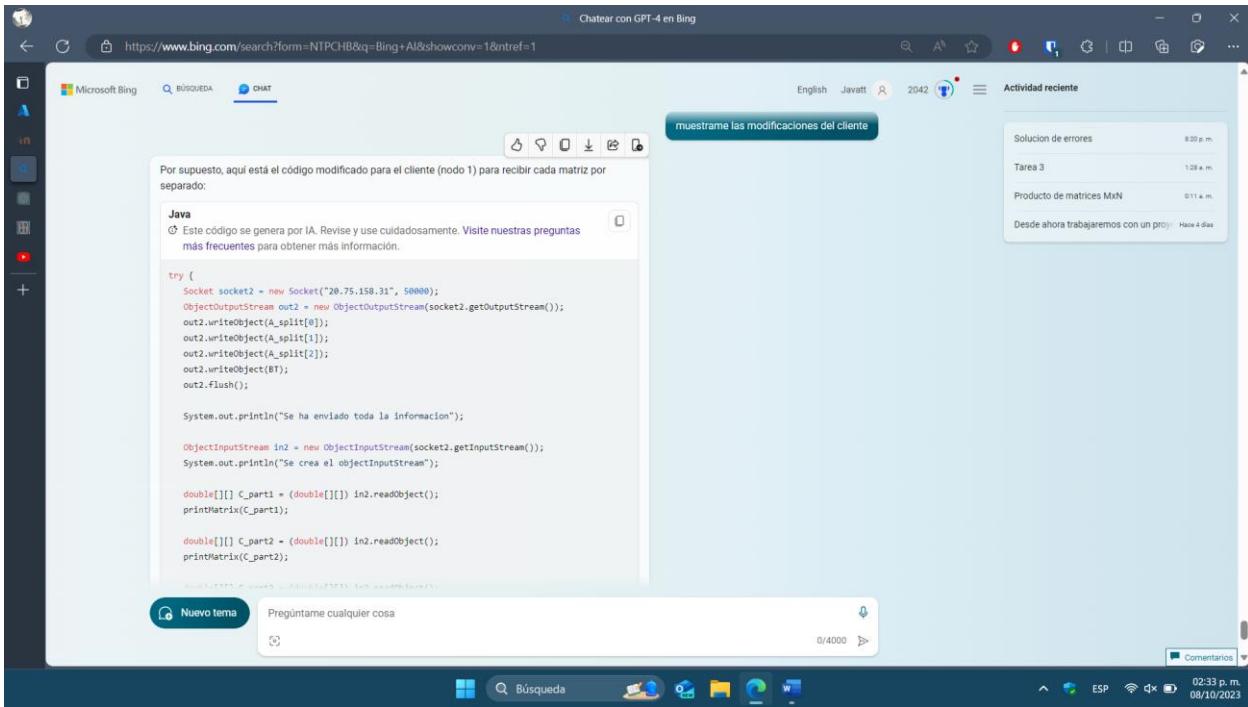
Este código se genera por IA. Revise y use cuidadosamente. Visite nuestras preguntas más frecuentes para obtener más información.

```
public void run() {
    try {
        ObjectInputStream in = new ObjectInputStream(socket.getInputStream());
        double[][] A_part1 = (double[][]) in.readObject();
        printMatrix(A_part1);
        double[][] A_part2 = (double[][]) in.readObject();
        printMatrix(A_part2);
        double[][] A_part3 = (double[][]) in.readObject();
        printMatrix(A_part3);
        double[][] BT = (double[][]) in.readObject();
        printMatrix(BT);

        double[][] C_part1 = multiply(A_part1, BT);
        printMatrix(C_part1);
        double[][] C_part2 = multiply(A_part2, BT);
        printMatrix(C_part2);
        double[][] C_part3 = multiply(A_part3, BT);
        printMatrix(C_part3);
    }
}
```

Below the code, there is a message input field with the placeholder "Pregúntame cualquier cosa" and a character limit of "0/4000". The bottom right corner of the window shows the date and time as "02:32 p. m. 08/10/2023".

Y en el cliente de igual forma se lee cada multiplicación por separado.

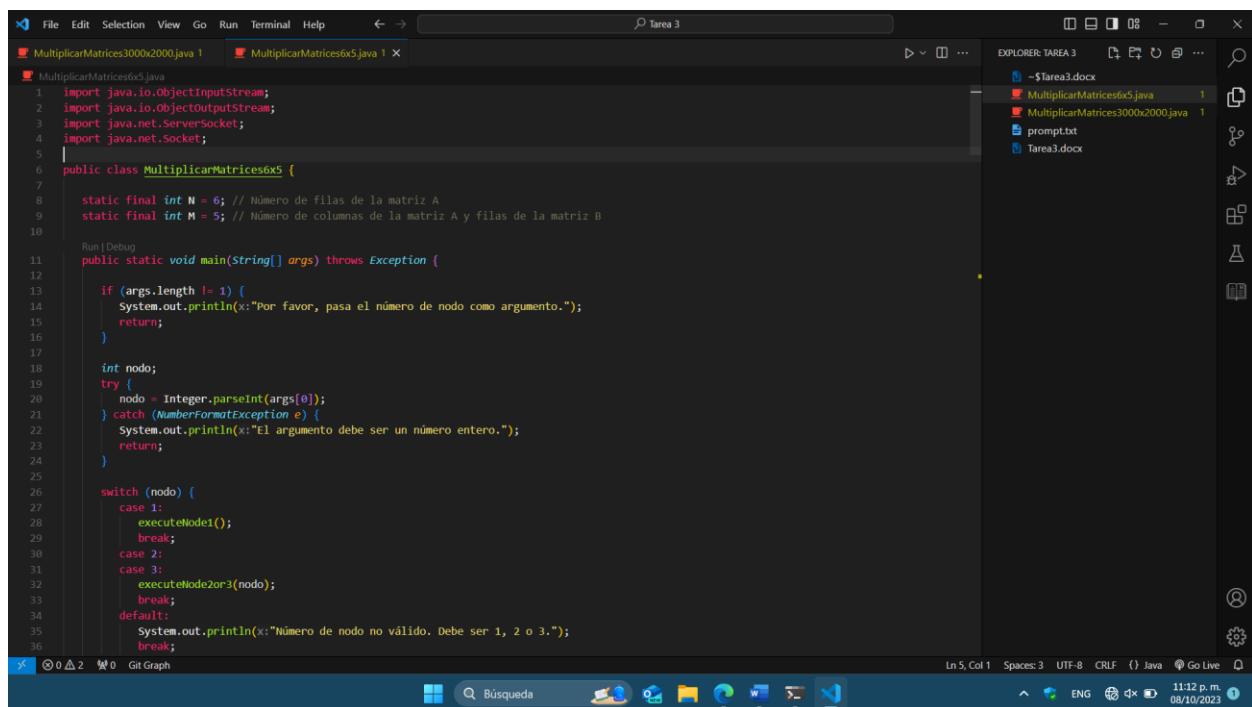


Con esto finalizado, ya obtenemos una estructura bastante sólida sobre el problema que queremos resolver, como se pudo observar esta IA si bien es cierto que en ciertas partes fue bastante precisa, tuvo ciertos errores al entender que es lo que queríamos específicamente por lo que nuevamente toca tomar lo que nos proporciona y hacer las modificaciones específicas para que todo funcione.

Códigos

Finalmente mostraremos los códigos completos, cabe aclarar que en la ejecución de estos en las maquinas virtuales se crearon dos versiones del mismo código para más comodidad al momento de hacer pruebas, estas son “MultiplicarMatrices6x5.java” y “MultiplicarMatrices3000x2000” lo único que cambia el uno con el otro es la omisión ciertas líneas de código que se enfocaban a imprimir las matrices y el tamaño de las matrices.

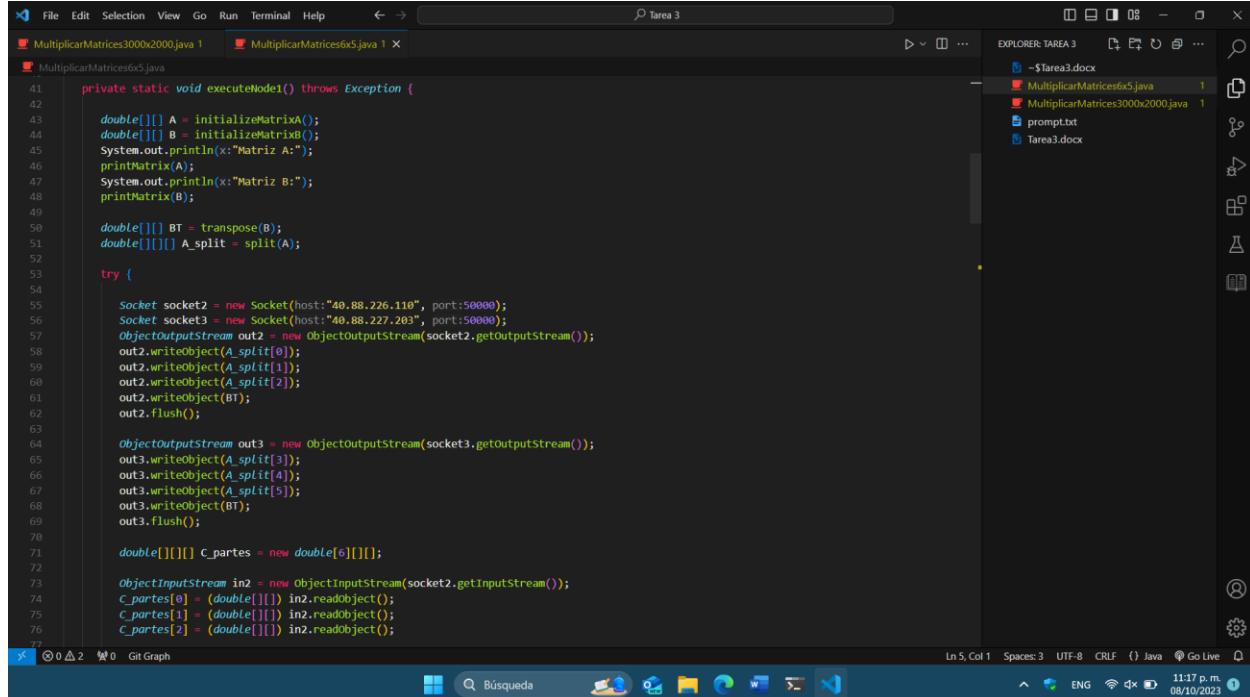
Primeramente, creamos dos variables estáticas, las cuales servirán para definir la longitud de las matrices (estos valores cambian según la prueba, ya sea 6 y 5 o 3000 y 2000). Seguido de esto tenemos el main, en el cual se verifica que el usuario haya puesto un parámetro al ejecutar el programa y que sea un numero este parámetro. Después se analiza que nodo fue el que el usuario digito y se manda a llamar la función respectiva para cada nodo.



```
MultiplicarMatrices6x5.java
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.ServerSocket;
import java.net.Socket;
public class MultiplicarMatrices6x5 {
    static final int N = 6; // Número de filas de la matriz A
    static final int M = 5; // Número de columnas de la matriz A y filas de la matriz B
    public static void main(String[] args) throws Exception {
        if (args.length != 1) {
            System.out.println("Por favor, pasa el número de nodo como argumento.");
            return;
        }
        int nodo;
        try {
            nodo = Integer.parseInt(args[0]);
        } catch (NumberFormatException e) {
            System.out.println("El argumento debe ser un número entero.");
            return;
        }
        switch (nodo) {
            case 1:
                executeNode1();
                break;
            case 2:
            case 3:
                executeNode2or3(nodo);
                break;
            default:
                System.out.println("Número de nodo no válido. Debe ser 1, 2 o 3.");
                break;
    }
}
}

MultiplicarMatrices3000x2000.java
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.ServerSocket;
import java.net.Socket;
public class MultiplicarMatrices3000x2000 {
    static final int N = 3000; // Número de filas de la matriz A
    static final int M = 2000; // Número de columnas de la matriz A y filas de la matriz B
    public static void main(String[] args) throws Exception {
        if (args.length != 1) {
            System.out.println("Por favor, pasa el número de nodo como argumento.");
            return;
        }
        int nodo;
        try {
            nodo = Integer.parseInt(args[0]);
        } catch (NumberFormatException e) {
            System.out.println("El argumento debe ser un número entero.");
            return;
        }
        switch (nodo) {
            case 1:
                executeNode1();
                break;
            case 2:
            case 3:
                executeNode2or3(nodo);
                break;
            default:
                System.out.println("Número de nodo no válido. Debe ser 1, 2 o 3.");
                break;
    }
}
}
```

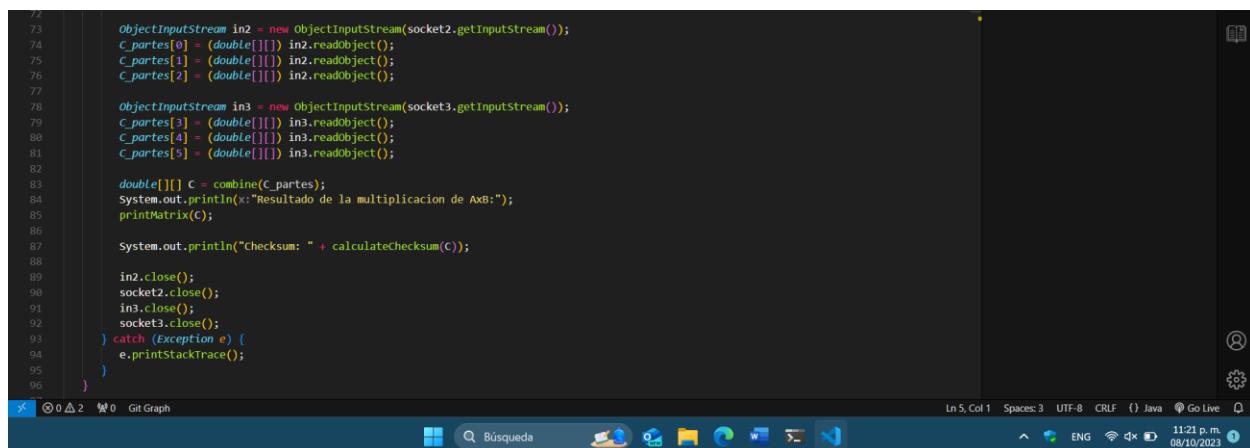
En el código del nodo 1 se inicializan las matrices, se obtiene la transpuesta de B, se divide en partes A, se abre la conexión a cada servidor y se manda las partes que nos permitan generar de C0 a C18 al nodo 2 y de C19 a C36 al nodo 3. Se reciben la multiplicación de matrices de cada nodo, se junta y se imprime el resultado.



```

1  private static void executeNode1() throws Exception {
2
3      double[][] A = initializeMatrixA();
4      double[][] B = initializeMatrixB();
5      System.out.println("Matriz A:");
6      printMatrix(A);
7      System.out.println("Matriz B:");
8      printMatrix(B);
9
10     double[][] BT = transpose(B);
11     double[][][] A_split = split(A);
12
13     try {
14
15         Socket socket2 = new Socket(host:"40.88.226.110", port:50000);
16         Socket socket3 = new Socket(host:"40.88.227.203", port:50000);
17         ObjectOutputStream out2 = new ObjectOutputStream(socket2.getOutputStream());
18         out2.writeObject(A_split[0]);
19         out2.writeObject(A_split[1]);
20         out2.writeObject(A_split[2]);
21         out2.writeObject(BT);
22         out2.flush();
23
24         ObjectOutputStream out3 = new ObjectOutputStream(socket3.getOutputStream());
25         out3.writeObject(A_split[3]);
26         out3.writeObject(A_split[4]);
27         out3.writeObject(A_split[5]);
28         out3.writeObject(BT);
29         out3.flush();
30
31         double[][][] C_partes = new double[6][][];
32
33         ObjectInputStream in2 = new ObjectInputStream(socket2.getInputStream());
34         C_partes[0] = (double[][]) in2.readObject();
35         C_partes[1] = (double[][]) in2.readObject();
36         C_partes[2] = (double[][]) in2.readObject();
37
38         ObjectInputStream in3 = new ObjectInputStream(socket3.getInputStream());
39         C_partes[3] = (double[][]) in3.readObject();
40         C_partes[4] = (double[][]) in3.readObject();
41         C_partes[5] = (double[][]) in3.readObject();
42
43         double[][] C = combine(C_partes);
44         System.out.println("Resultado de la multiplicacion de AxB:");
45         printMatrix(C);
46
47         System.out.println("Checksum: " + calculateChecksum(C));
48
49         in2.close();
50         socket2.close();
51         in3.close();
52         socket3.close();
53     } catch (Exception e) {
54         e.printStackTrace();
55     }
56 }

```

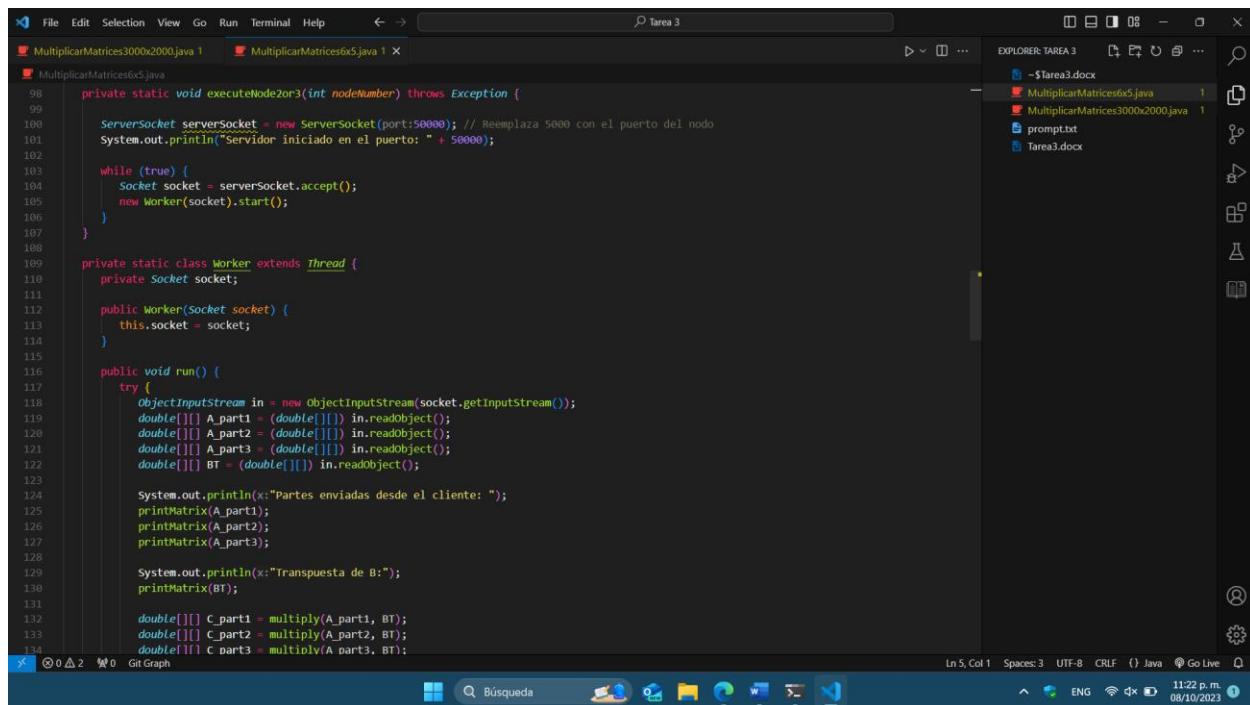


```

73
74     ObjectInputStream in2 = new ObjectInputStream(socket2.getInputStream());
75     C_partes[0] = (double[][]) in2.readObject();
76     C_partes[1] = (double[][]) in2.readObject();
77     C_partes[2] = (double[][]) in2.readObject();
78
79     ObjectInputStream in3 = new ObjectInputStream(socket3.getInputStream());
80     C_partes[3] = (double[][]) in3.readObject();
81     C_partes[4] = (double[][]) in3.readObject();
82     C_partes[5] = (double[][]) in3.readObject();
83
84     double[][] C = combine(C_partes);
85     System.out.println("Resultado de la multiplicacion de AxB:");
86     printMatrix(C);
87
88     System.out.println("Checksum: " + calculateChecksum(C));
89
90     in2.close();
91     socket2.close();
92     in3.close();
93     socket3.close();
94 } catch (Exception e) {
95     e.printStackTrace();
96 }

```

En el código del nodo 2 y 3 se crea un servidor TCP multithread que esta escuchando peticiones y al momento de recibir una crea un hilo, dentro del hilo se reciben las partes de A que generen de C0 a C36 respectivamente y se calcula la multiplicación, paso seguido se envía el resultado de cada multiplicación al nodo 1. Es importante remarcar el uso de flush(), ya que sin poner esta función pueden ocurrir ciertos problemas de que no se envié toda la información correctamente o se continúe con el proceso sin enviar toda la información completa.



```

private static void executeNode2or3(int nodeNumber) throws Exception {
    ServerSocket serverSocket = new ServerSocket(port:5000); // Reemplaza 5000 con el puerto del nodo
    System.out.println("Servidor iniciado en el puerto: " + port);
    while (true) {
        Socket socket = serverSocket.accept();
        new Worker(socket).start();
    }
}

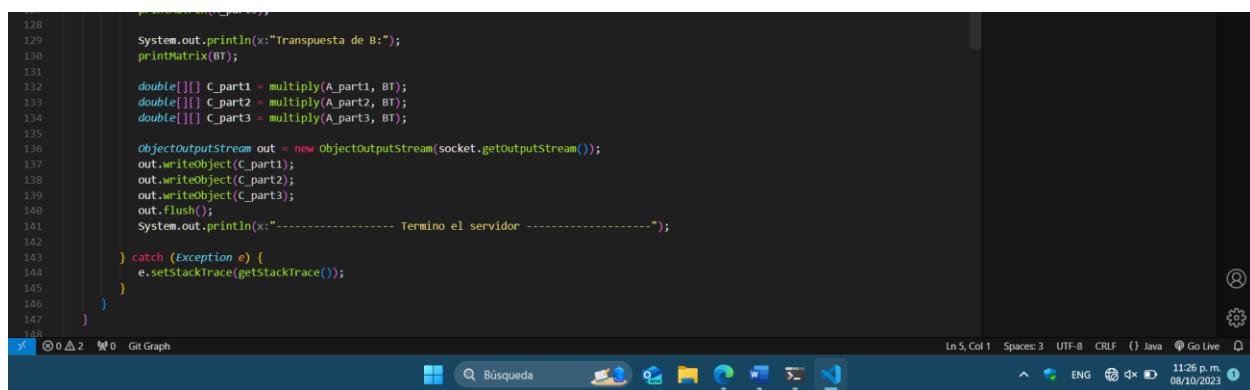
private static class Worker extends Thread {
    private Socket socket;
    public Worker(Socket socket) {
        this.socket = socket;
    }
    public void run() {
        try {
            ObjectInputStream in = new ObjectInputStream(socket.getInputStream());
            double[][] A_part1 = (double[][]) in.readObject();
            double[][] A_part2 = (double[][]) in.readObject();
            double[][] A_part3 = (double[][]) in.readObject();
            double[][] BT = (double[][]) in.readObject();

            System.out.println("Partes enviadas desde el cliente: ");
            printMatrix(A_part1);
            printMatrix(A_part2);
            printMatrix(A_part3);

            System.out.println("Transpuesta de B:");
            printMatrix(BT);

            double[][] C_part1 = multiply(A_part1, BT);
            double[][] C_part2 = multiply(A_part2, BT);
            double[][] C_part3 = multiply(A_part3, BT);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

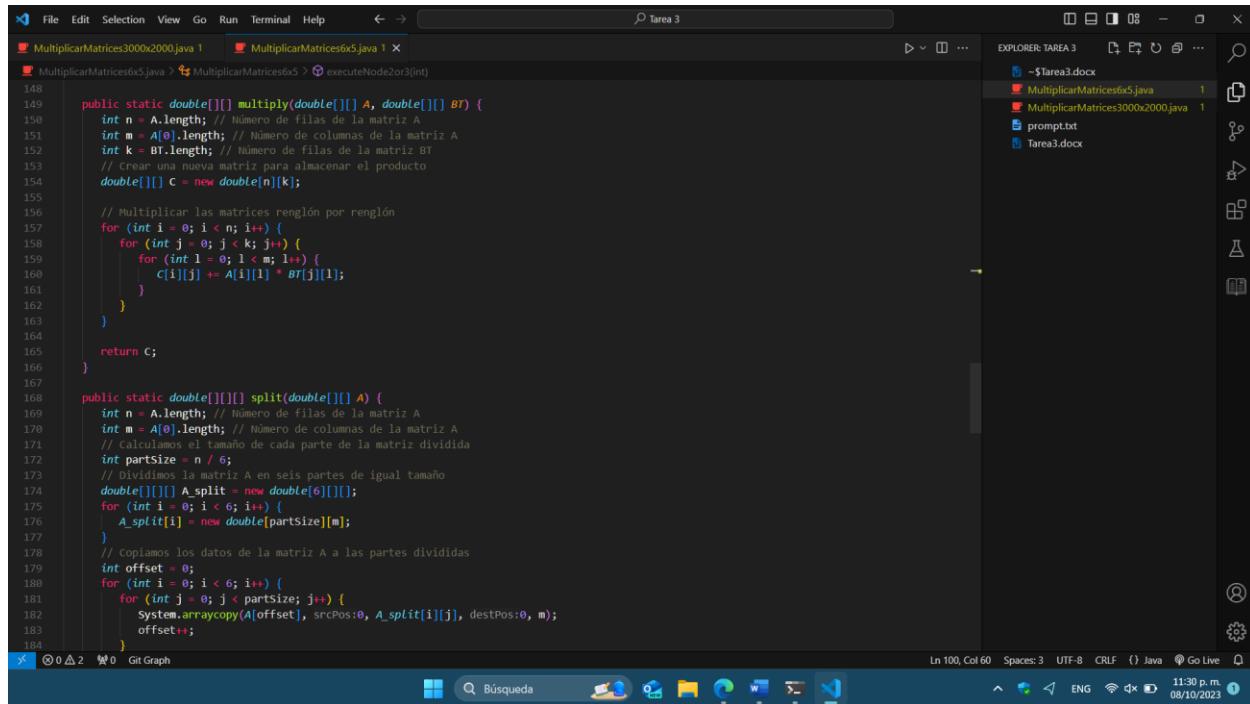


```

        ObjectOutputStream out = new ObjectOutputStream(socket.getOutputStream());
        out.writeObject(C_part1);
        out.writeObject(C_part2);
        out.writeObject(C_part3);
        out.flush();
        System.out.println("----- Termino el servidor -----");
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

En esta sección del código se definen todas las funciones que se usaron en todo el programa. Se define la función que multiplica renglón por renglón de una matriz A y una matriz Bt, de igual forma esta la función que se encarga dividir la matriz A en 6 partes iguales.

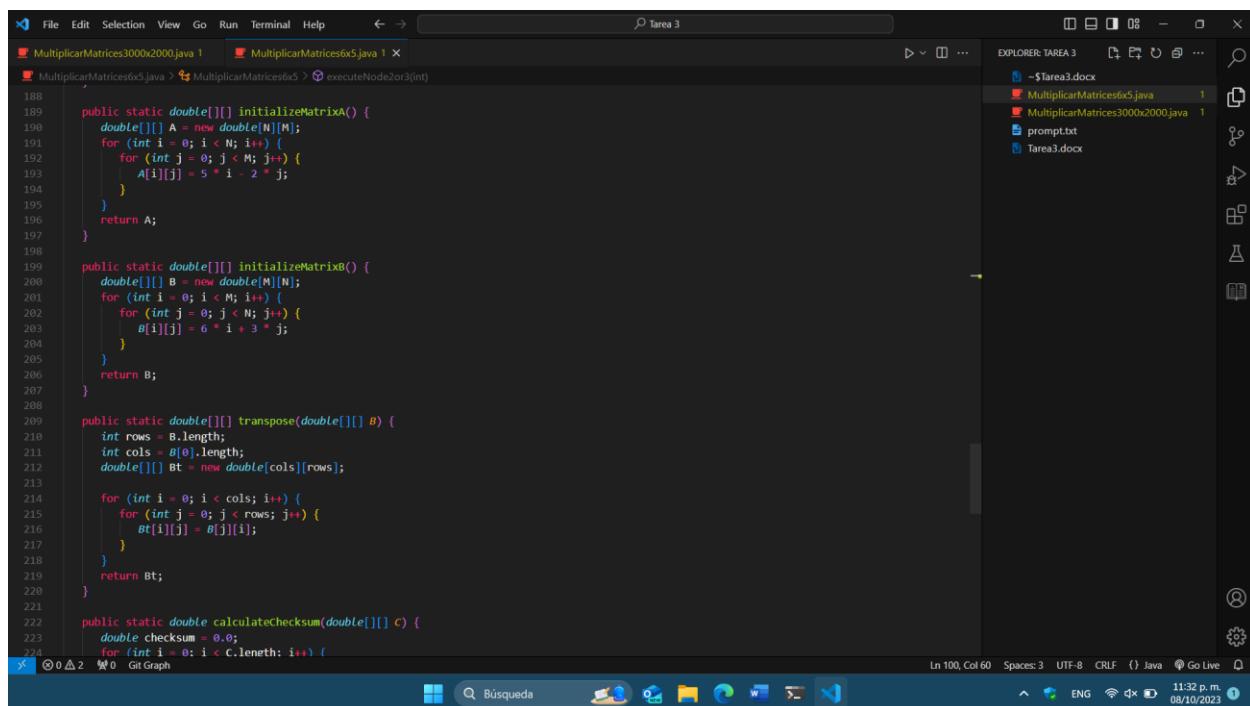


```

148 public static double[][] multiply(double[][] A, double[][] BT) {
149     int n = A.length; // Número de filas de la matriz A
150     int m = A[0].length; // Número de columnas de la matriz A
151     int k = BT.length; // Número de filas de la matriz BT
152     // Crear una nueva matriz para almacenar el producto
153     double[][] C = new double[n][k];
154
155     // Multiplicar las matrices renglón por renglón
156     for (int i = 0; i < n; i++) {
157         for (int j = 0; j < k; j++) {
158             for (int l = 0; l < m; l++) {
159                 C[i][j] += A[i][l] * BT[j][l];
160             }
161         }
162     }
163
164     return C;
165 }
166
167 public static double[][][] split(double[][][] A) {
168     int n = A.length; // Número de filas de la matriz A
169     int m = A[0].length; // Número de columnas de la matriz A
170     // Calculamos el tamaño de cada parte de la matriz dividida
171     int partSize = n / 6;
172     // Dividimos la matriz A en seis partes de igual tamaño
173     double[][][] A_split = new double[6][1][];
174     for (int i = 0; i < 6; i++) {
175         A_split[i] = new double[partSize][m];
176     }
177     // Copiamos los datos de la matriz A a las partes divididas
178     int offset = 0;
179     for (int i = 0; i < 6; i++) {
180         for (int j = 0; j < partSize; j++) {
181             System.arraycopy(A[offset], srcPos:0, A_split[i][j], destPos:0, m);
182             offset++;
183         }
184     }
185 }

```

En esta parte tenemos las definiciones de cada matriz de la forma en la que lo dicta el texto y de igual forma la función que nos permite transponer la matriz B.



```

188 public static double[][] initializeMatrixA() {
189     double[][] A = new double[M][M];
190     for (int i = 0; i < N; i++) {
191         for (int j = 0; j < M; j++) {
192             A[i][j] = 5 * i - 2 * j;
193         }
194     }
195     return A;
196 }
197
198 public static double[][] initializeMatrixB() {
199     double[][] B = new double[M][N];
200     for (int i = 0; i < M; i++) {
201         for (int j = 0; j < N; j++) {
202             B[i][j] = 6 * i + 3 * j;
203         }
204     }
205     return B;
206 }
207
208 public static double[][] transpose(double[][][] B) {
209     int rows = B.length;
210     int cols = B[0].length;
211     double[][] BT = new double[cols][rows];
212
213     for (int i = 0; i < cols; i++) {
214         for (int j = 0; j < rows; j++) {
215             BT[i][j] = B[j][i];
216         }
217     }
218     return BT;
219 }
220
221 public static double calculateChecksum(double[][][] C) {
222     double checksum = 0.0;
223     for (int i = 0; i < C.length; i++) {
224         for (int j = 0; j < C[i].length; j++) {
225             for (int k = 0; k < C[i][j].length; k++)
226                 checksum += C[i][j][k];
227         }
228     }
229     return checksum;
230 }

```

Por último, tenemos la función con la que calculamos el checksum, la función que nos permite combinar las partes de C y una función que nos imprime en pantalla las matrices.

The screenshot shows a Java IDE interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Tarea 3.
- Left Panel:** Shows two tabs: "MultiplicarMatrices3000x2000.java 1" and "MultiplicarMatrices6x5.java 1".
- Code Editor:** Displays Java code for matrix multiplication. The code includes methods for calculating checksums, combining matrices, and printing them. It uses double arrays for matrices and integer arrays for part sizes.
- Right Panel:** Explorer pane titled "EXPLORER: TAREA 3" showing files: "Tarea3.docx", "MultiplicarMatrices6x5.java", "MultiplicarMatrices3000x2000.java", "prompt.txt", and "Tarea3.docx".
- Bottom Status Bar:** Labeled "Ln 100, Col 60", "Spaces: 3", "UTF-8", "CRLF", "Java", and "Go Live".
- Bottom Icons:** Includes icons for search, file operations, and system status.

Ejecución y pruebas

Para poder iniciar con las pruebas previamente hemos instalado java versión 11.0.

```
javier@T3-2019630039-1:~$ java --version
openjdk 11.0.20.1 2023-08-24
OpenJDK Runtime Environment (build 11.0.20.1+1-post-Ubuntu-0ubuntu120.04)
OpenJDK 64-Bit Server VM (build 11.0.20.1+1-post-Ubuntu-0ubuntu120.04, mixed mode, sharing)
javier@T3-2019630039-1:~$ |
```

```
javier@T3-2019630039-2:~$ java --version
openjdk 11.0.20.1 2023-08-24
OpenJDK Runtime Environment (build 11.0.20.1+1-post-Ubuntu-0ubuntu120.04)
OpenJDK 64-Bit Server VM (build 11.0.20.1+1-post-Ubuntu-0ubuntu120.04, mixed mode, sharing)
javier@T3-2019630039-2:~$ |
```

```
javier@T3-2019630039-3:~$ java --version
openjdk 11.0.20.1 2023-08-24
OpenJDK Runtime Environment (build 11.0.20.1+1-post-Ubuntu-0ubuntu120.04)
OpenJDK 64-Bit Server VM (build 11.0.20.1+1-post-Ubuntu-0ubuntu120.04, mixed mode, sharing)
javier@T3-2019630039-3:~$ |
```

Y también deberemos precargar los archivos, como mencionamos, hemos puesto dos archivos en cada máquina para más practicidad y no tener que estar comentando líneas de código en cada máquina cada que se quiera hacer una prueba.

```
javier@T3-2019630039-1:~$ ls
'MultiplicarMatrices3000x2000$Worker.class'
MultiplicarMatrices3000x2000.class
MultiplicarMatrices3000x2000.java
'MultiplicarMatrices6x5$Worker.class'
MultiplicarMatrices6x5.class
MultiplicarMatrices6x5.java
javier@T3-2019630039-1:~$ |
```

```
javier@T3-2019630039-2:~$ ls
'MultiplicarMatrices3000x2000$Worker.class'
MultiplicarMatrices3000x2000.class
MultiplicarMatrices3000x2000.java
'MultiplicarMatrices6x5$Worker.class'
MultiplicarMatrices6x5.class
MultiplicarMatrices6x5.java
javier@T3-2019630039-2:~$ |
```

```
javier@T3-2019630039-3:~$ ls
'MultiplicarMatrices3000x2000$Worker.class'
MultiplicarMatrices3000x2000.class
MultiplicarMatrices3000x2000.java
'MultiplicarMatrices6x5$Worker.class'
MultiplicarMatrices6x5.class
MultiplicarMatrices6x5.java
javier@T3-2019630039-3:~$ |
```

En nuestro código, configurar las IPs de los nodos en el puerto 50000 (ambos deben tener los puertos abiertos).

```
Socket socket2 = new Socket(host:"40.88.226.110", port:50000);
Socket socket3 = new Socket(host:"40.88.227.203", port:50000);
ObjectOutputStream out2 = new ObjectOutputStream(socket2.getOutputStream());
```

Prueba 1. Matriz de 6x5

```
javier@T3-2019630039-1:~$ java MultiplicarMatrices6x5 1
Matriz A:
8.0 12.0 -4.0 -6.0 -8.0
5.0 3.0 1.0 -1.0 -3.0
10.0 8.0 6.0 4.0 2.0
15.0 13.0 11.0 9.0 7.0
20.0 18.0 16.0 14.0 12.0
25.0 23.0 21.0 19.0 17.0

Matriz B:
8.0 3.0 6.0 9.0 12.0 15.0
6.0 9.0 12.0 15.0 18.0 21.0
12.0 15.0 18.0 21.0 24.0 27.0
18.0 21.0 24.0 27.0 30.0 33.0
24.0 27.0 30.0 33.0 36.0 39.0

Resultado de la multiplicacion de AxB:
-360.0 -420.0 -480.0 -540.0 -600.0 -660.0
-40.0 -45.0 -30.0 -15.0 8.0 15.0
240.0 330.0 420.0 510.0 600.0 690.0
540.0 705.0 870.0 1035.0 1290.0 1365.0
840.0 1080.0 1320.0 1560.0 1800.0 2040.0
1140.0 1455.0 1770.0 2085.0 2400.0 2715.0

Checksum: 25515.8
javier@T3-2019630039-1:~$ |
```

javier@T3-2019630039-2:~\$ java MultiplicarMatrices6x5 2
Servidor iniciado en el puerto: 50000
Partes enviadas desde el cliente:
0.0 -2.0 -4.0 -6.0 -8.0

5.0 3.0 1.0 -1.0 -3.0

10.0 8.0 6.0 4.0 2.0

Transpuesta de B:
0.0 6.0 12.0 18.0 24.0
3.0 9.0 15.0 21.0 27.0
6.0 12.0 18.0 24.0 30.0
9.0 15.0 21.0 27.0 33.0
12.0 18.0 24.0 30.0 36.0
15.0 21.0 27.0 33.0 39.0

Termino el servidor

```
javier@T3-2019630039-3:~$ java MultiplicarMatrices6x5 3  
Servidor iniciado en el puerto: 50000  
Partes enviadas desde el cliente:  
15.0 13.0 11.0 9.0 7.0  
  
20.0 18.0 16.0 14.0 12.0  
  
25.0 23.0 21.0 19.0 17.0  
  
Transpuesta de B:  
0.0 6.0 12.0 18.0 24.0  
3.0 9.0 15.0 21.0 27.0  
6.0 12.0 18.0 24.0 30.0  
9.0 15.0 21.0 27.0 33.0  
12.0 18.0 24.0 30.0 36.0  
15.0 21.0 27.0 33.0 39.0
```

Termino el servidor

Para esta prueba se inicializo una matriz de 6x5 (A) y una de 5x6 (B):

```
javier@T3-2019630039-1:~/java MultiplicarMatrices6x5 1
Matriz A:
0.0 -2.0 -4.0 -6.0 -8.0
5.0 3.0 1.0 -1.0 -3.0
10.0 8.0 6.0 4.0 2.0
15.0 13.0 11.0 9.0 7.0
20.0 18.0 16.0 14.0 12.0
25.0 23.0 21.0 19.0 17.0

Matriz B:
0.0 3.0 6.0 9.0 12.0 15.0
6.0 9.0 12.0 15.0 18.0 21.0
12.0 15.0 18.0 21.0 24.0 27.0
18.0 21.0 24.0 27.0 30.0 33.0
24.0 27.0 30.0 33.0 36.0 39.0
```

Usaremos una página web como ayuda para obtener la multiplicación de estas dos matrices.

The screenshot shows a web-based matrix calculator interface. At the top, there are two input fields for matrices A and B, each containing a 6x6 grid of numerical values. Below these are several operation buttons: **A × B** (highlighted in blue), **A + B**, and **A - B**. To the right of the matrices are buttons for **Determinante** and **Matriz Inversa**. Further down are buttons for **Matriz Transpuesta**, **Multiplicar por** (with a dropdown set to 2), **Matriz Triangular**, **Matriz Diagonal**, **Matriz elevada a** (with a dropdown set to 2), and **Factorización LU**. On the left side, there's a sidebar with links to various matrix operations like **Operaciones con matrices**, **Solución de Sistemas de Ecuaciones Lineales**, **Calculadora de determinantes**, **Cálculo de valores propios y vectores propios**, and **Teoría necesaria**. At the bottom, there's a section for entering expressions like **2A+3B**, a checkbox for **Mostrar números decimales**, and a large text area showing the result of the multiplication. The result matrix is highlighted with a red box.

El resultado que devuelven los nodos 2 y 3 y que después juntamos es el siguiente:

$\begin{pmatrix} 0.0 & -2.0 & -4.0 & -6.0 & -8.0 \\ 5.0 & 3.0 & 1.0 & -1.0 & -3.0 \\ 10.0 & 8.0 & 6.0 & 4.0 & 2.0 \\ 15.0 & 13.0 & 11.0 & 9.0 & 7.0 \\ 20.0 & 18.0 & 16.0 & 14.0 & 12.0 \\ 25.0 & 23.0 & 21.0 & 19.0 & 17.0 \end{pmatrix}$	$\begin{pmatrix} 0.0 & 3.0 & 6.0 & 9.0 & 12.0 & 15.0 \\ 6.0 & 9.0 & 12.0 & 15.0 & 18.0 & 21.0 \\ 12.0 & 15.0 & 18.0 & 21.0 & 24.0 & 27.0 \\ 18.0 & 21.0 & 24.0 & 27.0 & 30.0 & 33.0 \\ 24.0 & 27.0 & 30.0 & 33.0 & 36.0 & 39.0 \\ 30.0 & 33.0 & 36.0 & 39.0 & 42.0 & 45.0 \end{pmatrix}$	$\begin{pmatrix} -360 & -420 & -480 & -540 & -600 & -660 \\ -60 & -45 & -30 & -15 & 0 & 15 \\ 240 & 330 & 420 & 510 & 600 & 690 \\ 540 & 705 & 870 & 1035 & 1200 & 1365 \\ 840 & 1080 & 1320 & 1560 & 1800 & 2040 \\ 1140 & 1455 & 1770 & 2085 & 2400 & 2715 \end{pmatrix}$ $\begin{pmatrix} 0.0 & -2.0 & -4.0 & -6.0 & -8.0 \\ 5.0 & 3.0 & 1.0 & -1.0 & -3.0 \\ 10.0 & 8.0 & 6.0 & 4.0 & 2.0 \\ 15.0 & 13.0 & 11.0 & 9.0 & 7.0 \\ 20.0 & 18.0 & 16.0 & 14.0 & 12.0 \\ 25.0 & 23.0 & 21.0 & 19.0 & 17.0 \end{pmatrix} \begin{pmatrix} 0.0 & 3.0 & 6.0 & 9.0 & 12.0 & 15.0 \\ 6.0 & 9.0 & 12.0 & 15.0 & 18.0 & 21.0 \\ 12.0 & 15.0 & 18.0 & 21.0 & 24.0 & 27.0 \\ 18.0 & 21.0 & 24.0 & 27.0 & 30.0 & 33.0 \\ 24.0 & 27.0 & 30.0 & 33.0 & 36.0 & 39.0 \\ 30.0 & 33.0 & 36.0 & 39.0 & 42.0 & 45.0 \end{pmatrix} = \begin{pmatrix} -360 & -420 & -480 & -540 & -600 & -660 \\ -60 & -45 & -30 & -15 & 0 & 15 \\ 240 & 330 & 420 & 510 & 600 & 690 \\ 540 & 705 & 870 & 1035 & 1200 & 1365 \\ 840 & 1080 & 1320 & 1560 & 1800 & 2040 \\ 1140 & 1455 & 1770 & 2085 & 2400 & 2715 \end{pmatrix}$
--	--	--

Resultado de la multiplicación de AxB:

```
-360.0 -420.0 -480.0 -540.0 -600.0 -660.0
-60.0 -45.0 -30.0 -15.0 0.0 15.0
240.0 330.0 420.0 510.0 600.0 690.0
540.0 705.0 870.0 1035.0 1200.0 1365.0
840.0 1080.0 1320.0 1560.0 1800.0 2040.0
1140.0 1455.0 1770.0 2085.0 2400.0 2715.0
```

Checksum: 25515.0

Si los comparamos, vemos que son prácticamente iguales, la pagina web realiza la multiplicación renglón por columna, pero nosotros lo hacemos renglón por renglón y a parte distribuimos el proceso a otros nodos, aun así, obtenemos el mismo resultado.

También, en las máquinas de los nodos 2 y 3, al ejecutar este código nos muestra que partes envió el cliente de A a cada nodo, de igual forma visualizamos Bt que es la transpuesta de B y un mensaje de cuando el servidor termina de contestar al nodo 1.

```
javier@T3-2019630039-2:~$ java MultiplicarMatrices6x5 2
Servidor iniciado en el puerto: 50000
Partes enviadas desde el cliente:
0.0 -2.0 -4.0 -6.0 -8.0

5.0 3.0 1.0 -1.0 -3.0

10.0 8.0 6.0 4.0 2.0

Transpuesta de B:
0.0 6.0 12.0 18.0 24.0
3.0 9.0 15.0 21.0 27.0
6.0 12.0 18.0 24.0 30.0
9.0 15.0 21.0 27.0 33.0
12.0 18.0 24.0 30.0 36.0
15.0 21.0 27.0 33.0 39.0

————— Termino el servidor ————
```

```
javier@T3-2019630039-3:~$ java MultiplicarMatrices6x5 3
Servidor iniciado en el puerto: 50000
Partes enviadas desde el cliente:
15.0 13.0 11.0 9.0 7.0

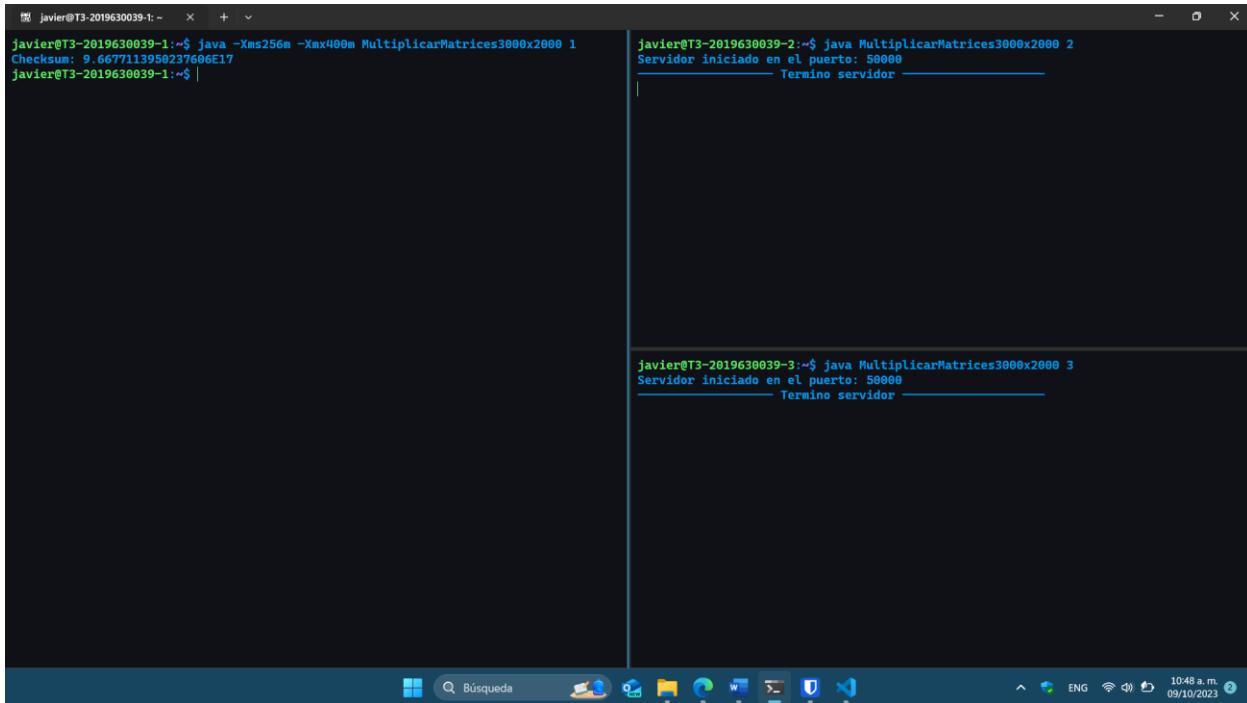
20.0 18.0 16.0 14.0 12.0

25.0 23.0 21.0 19.0 17.0

Transpuesta de B:
0.0 6.0 12.0 18.0 24.0
3.0 9.0 15.0 21.0 27.0
6.0 12.0 18.0 24.0 30.0
9.0 15.0 21.0 27.0 33.0
12.0 18.0 24.0 30.0 36.0
15.0 21.0 27.0 33.0 39.0

————— Termino el servidor ————
```

Prueba 2. Matriz de 3000x2000

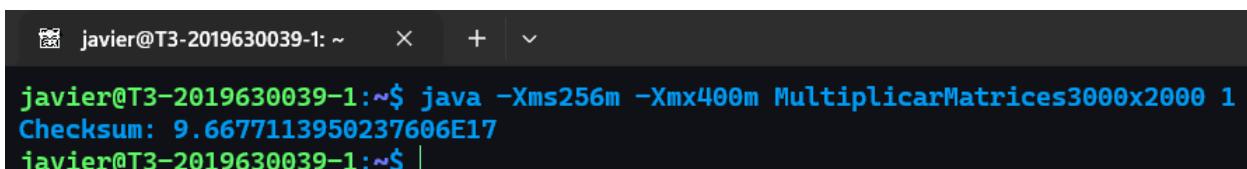


```
javier@T3-2019630039-1:~$ java -Xms256m -Xmx400m MultiplicarMatrices3000x2000 1
Checksum: 9.6677113950237606E17
javier@T3-2019630039-1:~$ |
```

```
javier@T3-2019630039-2:~$ java MultiplicarMatrices3000x2000 2
Servidor iniciado en el puerto: 50000
Termino servidor |
```

```
javier@T3-2019630039-3:~$ java MultiplicarMatrices3000x2000 3
Servidor iniciado en el puerto: 50000
Termino servidor |
```

Debido a que estamos usando matrices muy grandes y nuestro servidor es de 1gb de memoria, ocurría un error con la memoria heap, por lo que modificamos el comando para asignarle una memoria de mínimo 256MB y máximo 400MB para este caso, con esto ya no tenemos ningún problema al ejecutar el código del nodo 1 y obtenemos el siguiente checksum.



```
javier@T3-2019630039-1:~$ java -Xms256m -Xmx400m MultiplicarMatrices3000x2000 1
Checksum: 9.6677113950237606E17
javier@T3-2019630039-1:~$ |
```

Como mencionamos en este código de omiten la impresión de las matrices A,B y C en todos los nodos ya que si no le tomaría más tiempo al programa de terminar y no sería visualmente entendible.

Conclusión

En este programa se abordó un problema clásico en la computación, que es multiplicar matrices. Este problema ha sido objeto de intensa investigación a lo largo del tiempo, con el objetivo de encontrar formas más eficientes de abordarlo. La razón detrás de este interés radica en el hecho de que el manejo de arreglos bidimensionales puede consumir una cantidad significativa de memoria y tiempo de procesamiento.

En esta práctica, se optó por una solución más actual al delegar parte del procesamiento a otras máquinas, dividiendo las matrices para resolver el problema. Esta estrategia resultó ser altamente efectiva durante la ejecución del programa, ya que el tiempo de procesamiento se redujo considerablemente. Además, se aplicó una técnica de multiplicación de matrices que aprovechaba eficazmente el funcionamiento del búfer al multiplicar fila por fila de cada matriz.

Este enfoque no solo optimizó el rendimiento del programa, sino que también demostró la importancia de explorar soluciones creativas para desafíos computacionales comunes. En un mundo cada vez más orientado a la eficiencia y al procesamiento de datos a gran escala, la capacidad de abordar problemas clásicos de manera innovadora se convierte en una habilidad esencial en el campo de la informática y la computación.