



Instituto Politécnico Nacional



Escuela Superior de Cómputo

Diseño de Sistemas Distribuidos

Tarea 3 - Equipo 2

Docente:

Dr. Pineda Guerrero Carlos

Alumnos:

*Cazares Martínez Maximiliano
Chavarría Vázquez Luis Enrique
Cipriano Damián Sebastián*

Grupo: 4CV11

CDMX, 10 de marzo de 2022

Índice

Descripción del problema	3
Desarrollo.	6
Capturas de pantalla del código.	6
Especificaciones sobre el código y proceso de trabajo.	19
Conclusiones	20
Cazares Martínez Maximiliano	20
Chavarría Vázquez Luis Enrique	20
Cipriano Damián Sebastián.	20

Índice de imágenes

Ilustración 1 Creación de la máquina virtual.....	6
Ilustración 2 Creación de la máquina virtual, nombre de usuario y contraseña	7
Ilustración 3 Sección de los discos para la creación de la máquina virtual.	7
Ilustración 4 Creación de la máquina virtual sección de administración.	8
Ilustración 5 Sección de revisar y verificar para la creación de la máquina virtual.	9
Ilustración 6 Creación del proceso e implementación en curso.....	9
Ilustración 7 Proceso completado para la creación de la máquina virtual.....	10
Ilustración 8 Datos de la creación de la máquina virtual.	10
Ilustración 9 Verificación con el uso de la consola.....	11
Ilustración 10 Acceso de la terminal.	11
Ilustración 11 Actualizaciones dentro del sistema de la maquina virtual.....	12
Ilustración 12 Instalación del JRE.....	12
Ilustración 13 Instalación del JDK dentro de la máquina virtual.	13
Ilustración 14 Conexión dentro de la consola.....	13
Ilustración 15 Apreciación del código desde el editor de nano.....	14
Ilustración 16 Sección de redes de la máquina virtual.	14
Ilustración 17 Muestra de la creación de todas las máquina virtuales necesarias.....	15
Ilustración 18 Abrimos el archivo en nano para su posterior edición.	15
Ilustración 19 Modificación del archivo original para verificar que todo esta en funcionamiento.	16
Ilustración 20 Despliegue de todas las consolas de cada una de las máquinas virtuales.....	17
Ilustración 21 Ejecución del programa para los nodos 1,2,3 en el caso de N = 8.....	17
Ilustración 22 Ejecución final del código para el caso de N = 8.....	18
Ilustración 23 Ejecución del programa en el caso de N = 1000.....	18
Ilustración 24 Ejecución final del programa para el caso de N = 1000.....	19

Descripción del problema

En esta tarea cada alumno deberá desarrollar **un solo programa** en Java, el cual calculará el producto de dos matrices cuadradas en forma distribuida sobre cuatro nodos.

Sean A, B y C matrices cuadradas con elementos de tipo double, N renglones y N columnas, N par y $C = A \times B$.

Se deberá ejecutar dos casos:

1. $N=8$, desplegar las matrices A, B y C y el checksum de la matriz C.
2. $N=1000$, desplegar el checksum de la matriz C.

El checksum de la matriz C se calculará como la suma de todos elementos de la matriz. El checksum deberá ser de tipo double.

$$\text{checksum} = \sum C[i][j], i=0, \dots, N-1, j=0, \dots, N-1.$$

Se deberá inicializar las matrices de la siguiente manera:

$$A[i][j] = i + 5*j$$

$$B[i][j] = 5*i - j$$

Donde $A[i][j]$ y $B[i][j]$ son los elementos A_{ij} y B_{ij} respectivamente.

El programa deberá ser ejecutado en cuatro **máquinas virtuales con Ubuntu** (1 CPU, 1GB de RAM y disco HDD estándar) en cada máquina virtual se pasará como parámetro al programa el número de nodo, a saber: 0, 1, 2 y 3.

El nombre de cada máquina virtual **deberá** ser: "Tarea-3-" concatenando el número de equipo, un guion y el número de nodo, por ejemplo, si el equipo es el 12, entonces el nodo 0 deberá llamarse: Tarea-3-12-0, el nodo 1 deberá llamarse Tarea-3-12-1, y así sucesivamente. **No se admitirá la tarea** si los nodos no se nombran como se indicó anteriormente.

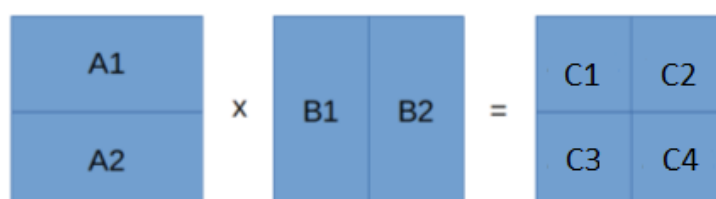
Recuerden que deben eliminar las máquinas virtuales cuando no las usen, con la finalidad de ahorrar el saldo de sus cuentas de Azure.

¿Cómo realizar la multiplicación de matrices en forma distribuida?

Suponga que divide la matriz A en las matrices A1 y A2. El tamaño de las matrices A1 y A2 es $N/2$ renglones y N columnas.

La matriz B se divide en las matrices B1 y B2. El tamaño de matrices B1 y B2 es N renglones y $N/2$ columnas.

Entonces la matriz $C = A \times B$ se compone de las matrices C1, C2, C3 y C4, tal como se muestra en la siguiente figura:



Donde:

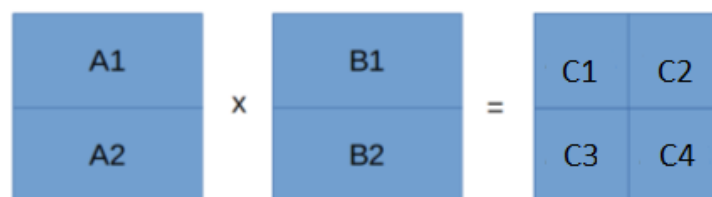
$$C1 = A1 \times B1$$

$$C2 = A1 \times B2$$

$$C3 = A2 \times B1$$

$$C4 = A2 \times B2$$

Debido a que las matrices se guardan en memoria por renglones, es más eficiente transponer la matriz B y dividirla de la siguiente manera:



En este caso el producto se realizará renglón por renglón, entonces:

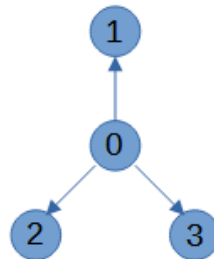
$$C1 = A1 \times B1$$

$$C2 = A1 \times B2$$

$$C3 = A2 \times B1$$

$$C4 = A2 \times B2$$

Ahora supongamos que tenemos cuatro nodos identificados con los números 0, 1, 2 y 3, el nodo 0 actuará como cliente y los nodos 1, 2 y 3 como servidores.



Para multiplicar las matrices A y B se deberá implementar los siguientes algoritmos:

Nodo 0:

1. Inicializar las matrices A y B.
2. Transponer la matriz B. Dejar la transpuesta en la misma matriz B.
3. Enviar la matriz A1 al nodo 1.
4. Enviar la matriz B1 al nodo 1.
5. Enviar la matriz A1 al nodo 2.
6. Enviar la matriz B2 al nodo 2.
7. Enviar la matriz A2 al nodo 3.

8. Enviar la matriz B1 al nodo 3.
9. Realizar el producto $C4=A2 \times B2$ (renglón por renglón).
10. Recibir la matriz C1 del nodo 1.
11. Recibir la matriz C2 del nodo 2.
12. Recibir la matriz C3 del nodo 3.
13. Calcular el checksum de la matriz C.
14. Desplegar el checksum de la matriz C.
15. Si $N=8$ entonces desplegar las matrices A, B y C

Nodo 1:

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

Nodo 2:

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

Nodo 3:

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

Notar que en el paso 3 del procedimiento que ejecuta en los nodos 1, 2 y 3, el producto de matrices (renglón por renglón) se debe realizar intercambiando los índices de la matriz B, tal como se hizo en el programa `MultiplicaMatriz_2.java`.

Se deberá subir a la plataforma el código fuente del programa desarrollado y un documento PDF con portada, las capturas de pantalla de la compilación y ejecución del programa. El archivo PDF deberá incluir una descripción de cada captura de pantalla y conclusiones.

El reporte deberá incluir **captura de pantalla de cada paso correspondiente a la creación de la primera máquina virtual** (nodo 0).

No se admitirá la tarea si no incluye todas las capturas de pantalla correspondientes a la creación de la primera máquina virtual (nodo 0).

La tarea **solo se admitirá** si el programa obtiene los resultados correctos para $N=8$ y para $N=1000$.

Valor de la tarea: 20% (1.4 puntos de la primera evaluación parcial)

Desarrollo.

Capturas de pantalla del código.

En la siguiente parte del reporte presentaremos las capturas de pantalla obtenidas durante las pruebas que hemos realizado.

En la imagen número 1, lo primero que estamos haciendo es crear nuestra máquina virtual, de modo tal que ingresamos los datos a la interfaz de Azure, de entre los cuales se encuentran algunos como el tipo de la suscripción, el nombre de la maquina virtual, la región a la que estamos adscritos, la disponibilidad, el tipo de seguridad del sistema, el sistema operativo o mejor dicho la imagen ISO con la que trabajaremos, que en este caso será Ubuntu y finalmente el tamaño de la misma máquina. Este proceso es fundamental al momento de crear nuestras máquinas ya que estamos definiendo los aspectos esenciales.

Crear una máquina virtual - Mic... x SSIS-4CV11: Tarea 3. Multiplic... x +

https://portal.azure.com/?Microsoft_Azure_Education_correlationId=102197db68a64b6dbff706b146a396bb&Microsoft_Azure_Education_newA4E=true&Microsoft_Azure_Education_as...

Microsoft Azure

Inicio > Máquinas virtuales >

Crear una máquina virtual

Datos básicos Discos Redes Administración Opciones avanzadas Etiquetas Revisar y crear

Cree una máquina virtual que ejecuta Linux o Windows. Seleccione una imagen de Azure Marketplace o use una imagen personalizada propia. Complete la pestaña Conceptos básicos y, después, use Revisar y crear para aprovisionar una máquina virtual con parámetros predeterminados o bien revise cada una de las pestañas para personalizar la configuración. [Más información](#)

Detalles del proyecto

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción *

Grupo de recursos * [Crear nuevo](#)

Detalles de instancia

Nombre de máquina virtual *

Región *

Opciones de disponibilidad

Tipo de seguridad

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

Instancia de Azure de acceso puntual ☐

Tamaño * [Ver todos los tamaños](#)

Cuenta de administrador

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

19°C Despejado 3:38 AM 3/8/2022

Ilustración 1 Creación de la máquina virtual

Para este debemos definir nuestra contraseña y el nombre del usuario, aunque también debe quedar clara la parte de si los puertos de entrada serán públicos, en esta sección debemos darle a que si se permitan los puertos; posteriormente debemos hacer la selección de SSH (22) para los puertos. Al tratarse de solamente una prueba no hay mayor problema de seguridad al dejar que las IPs puedan acceder. Queremos aclarar que la contraseña va de acuerdo con los nodos NodoXEquipo2 donde X es el número del nodo, por ejemplo, Nodo0Equipo2 para el Nodo 0 y así para cada uno, mientras que el nombre de los usuarios va dependiendo del número de los nodos, por ejemplo, en el caso del Nodo 0 es NodoCero y en el caso del uno es NodoUno.

Ilustración 2 Creación de la máquina virtual, nombre de usuario y contraseña

En la tercera imagen, nos vamos a la sección de discos en donde debemos seleccionar el HDD estándar, la opción de eliminar con el virtual machine o, mejor dicho, la maquina virtual; en la parte del cifrado debemos seleccionar el cifrado que esta predeterminado y básicamente con esto damos por concluida la parte de los discos, para poder continuar con el proceso de creación de nuestra máquina virtual.

Ilustración 3 Sección de los discos para la creación de la máquina virtual.

En esta imagen de la parte de abajo, ya una vez que hemos pasado por la sección de discos y redes, es importante estar en la sección de administración en donde deberemos verificar los datos que aparecen a continuación; esto de acuerdo con la información que revisamos debe estar de esta forma para su correcto funcionamiento.

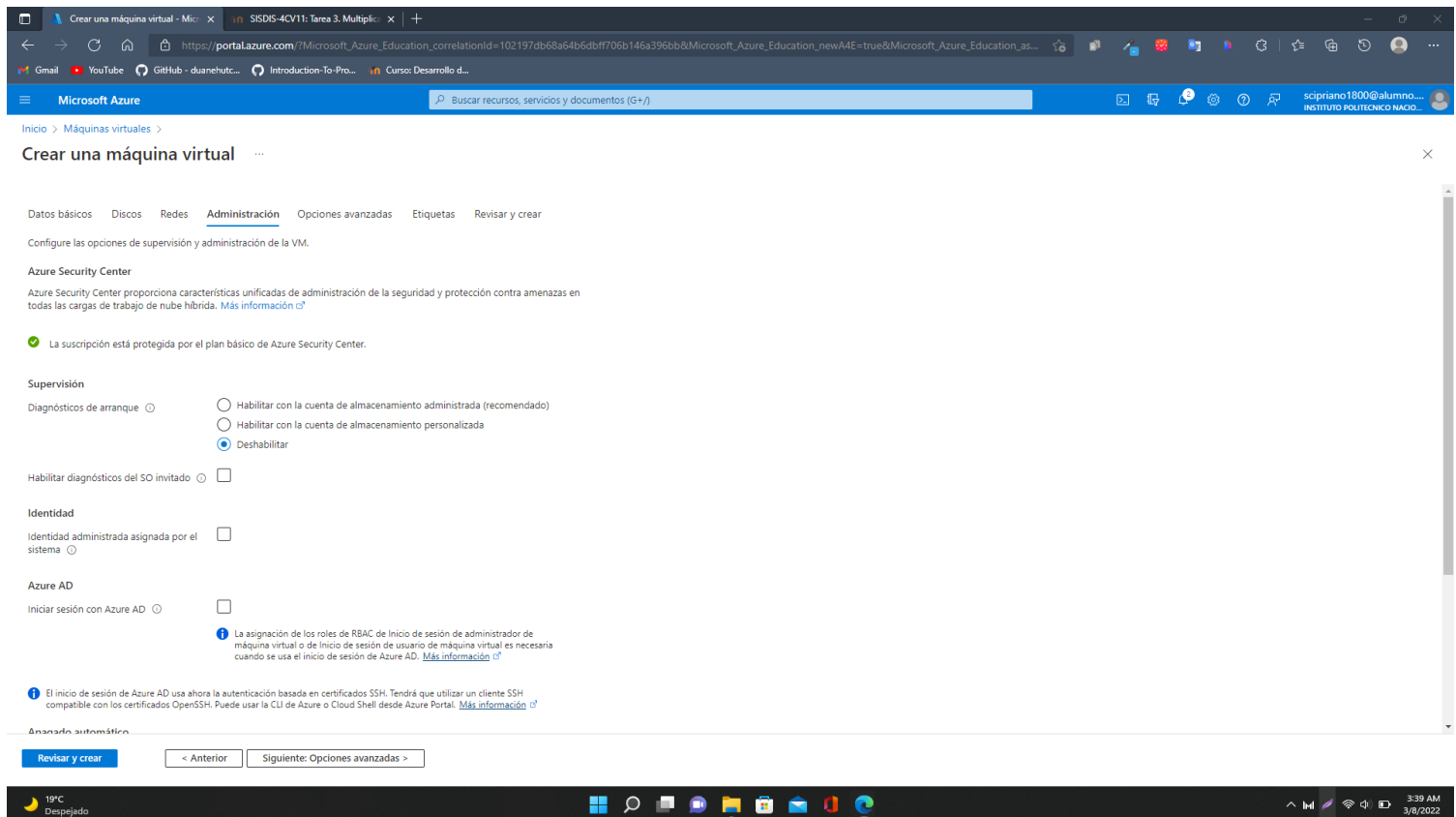


Ilustración 4 Creación de la máquina virtual sección de administración.

Ahora, debemos irnos directamente a la sección de revisar y crear en donde nos harán una especie de resumen de los datos de la máquina virtual y además de ellos podemos ver el costo que el uso de la máquina tendrá por hora de consumo, que, si bien no es muy elevado, debemos procurar mantenerla apagada para evitar gastar dinero del saldo que nos ha regalado la plataforma. Esto lo podemos ver en la imagen de abajo.

Por cuestiones del formato del documento la imagen quedó en la siguiente hoja.

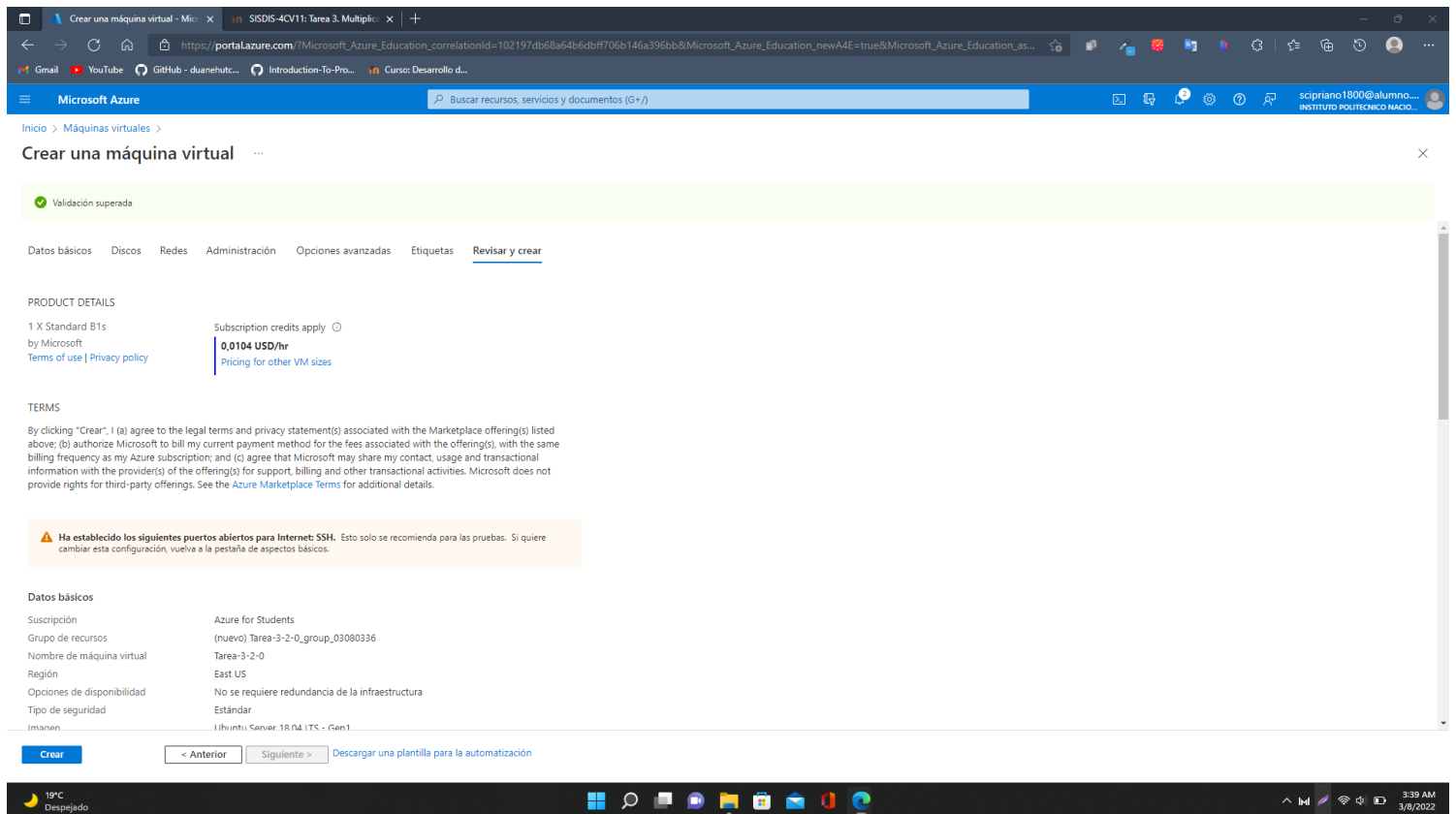


Ilustración 5 Sección de revisar y verificar para la creación de la máquina virtual.

En la imagen de la parte de abajo, vemos como el proceso ya se esta creando, de modo tal que vemos en la pantalla los múltiples detalles de la implementación, en la interfaz podemos ver las entradas, salidas y datos de la plantilla de la máquina virtual.

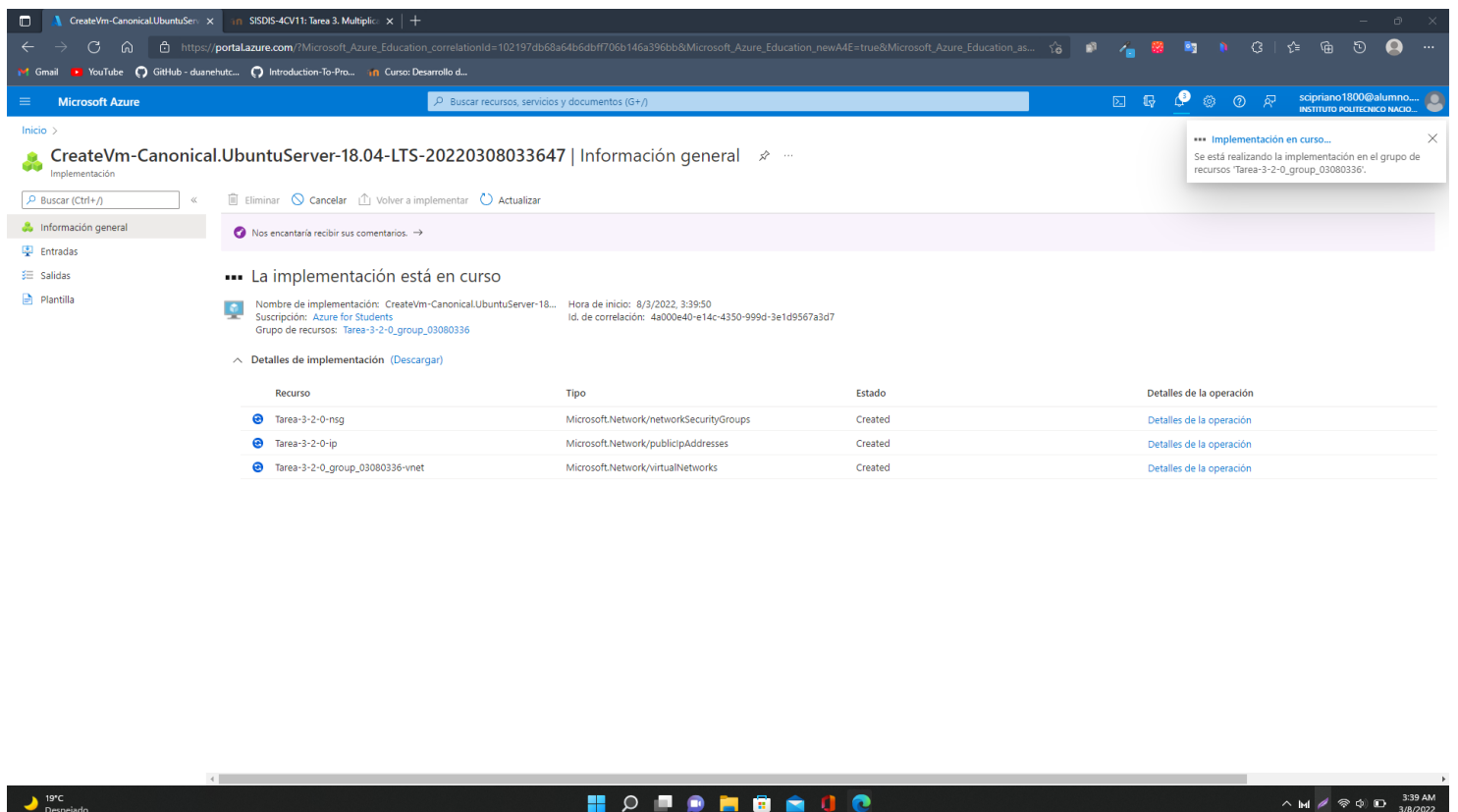


Ilustración 6 Creación del proceso e implementación en curso.

Una vez que se completo el proceso de la creación de nuestra máquina virtual hemos de poder ver la siguiente pantalla donde Azure nos indica que el proceso ha sido concluido de manera satisfactoria, podemos ver un resumen de los pasos que deben proceder para poder seguir con la implementación.

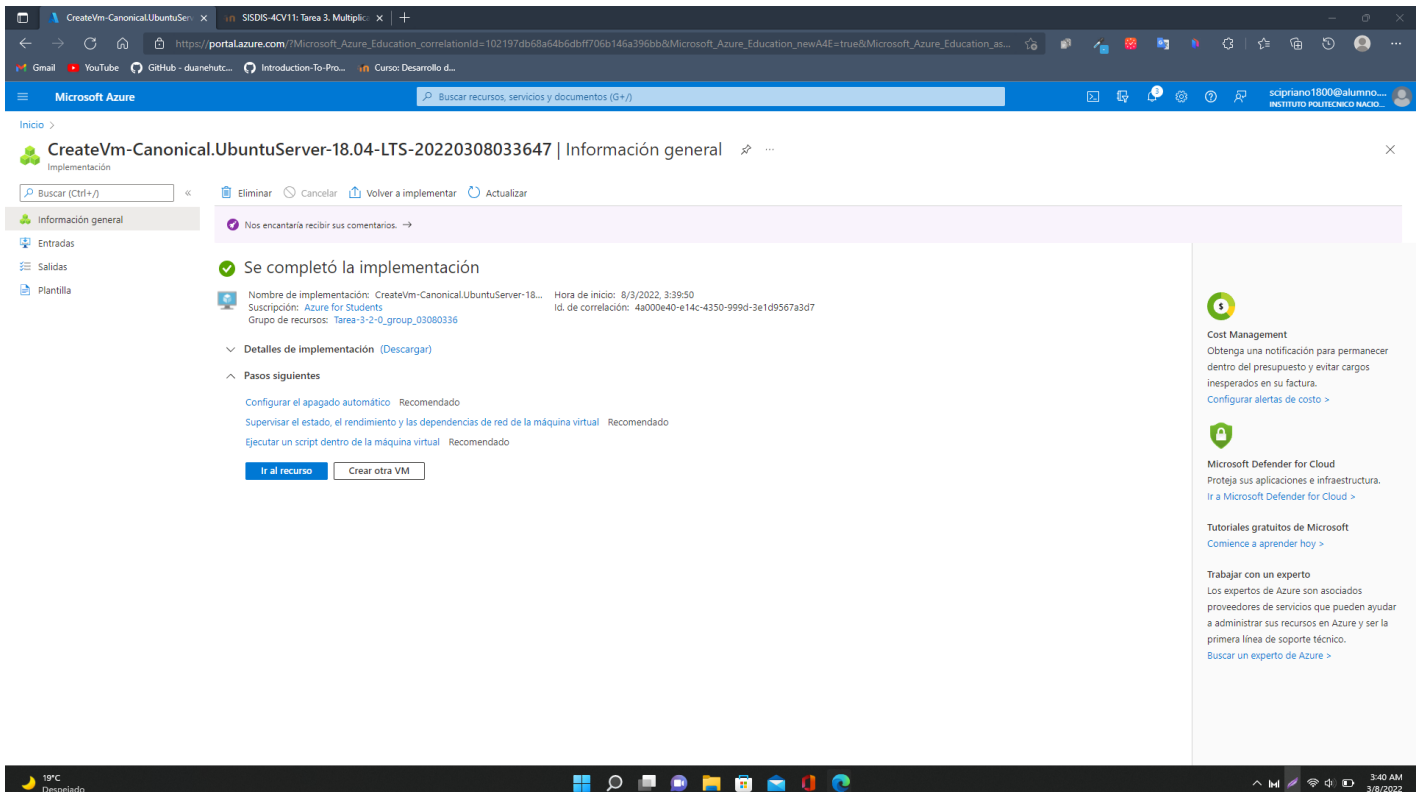


Ilustración 7 Proceso completado para la creación de la máquina virtual.

Podemos ver más de los datos de nuestra máquina virtual y también podemos ver como nuestra máquina virtual esta en el proceso de la ejecución, esto es imperante si queremos volver a ver información relevante que nos pueda ser de ayuda.

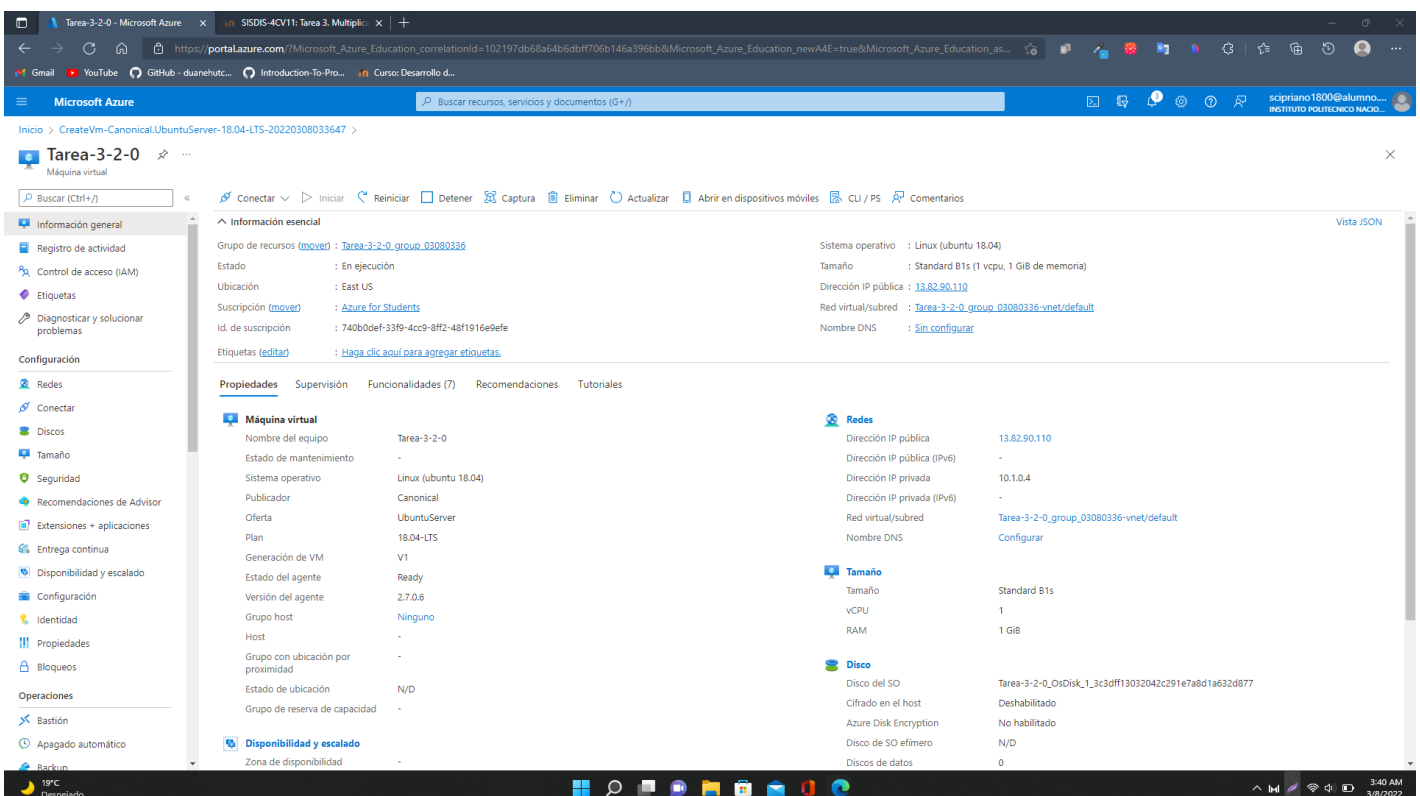


Ilustración 8 Datos de la creación de la máquina virtual.

Podemos ver en la imagen de la parte de abajo como nuestra consola para la máquina virtual ya puede ser accesada sin ningún tipo de problema y del mismo modo podemos comprobar su estado por si tenemos la duda del funcionamiento de la máquina virtual.

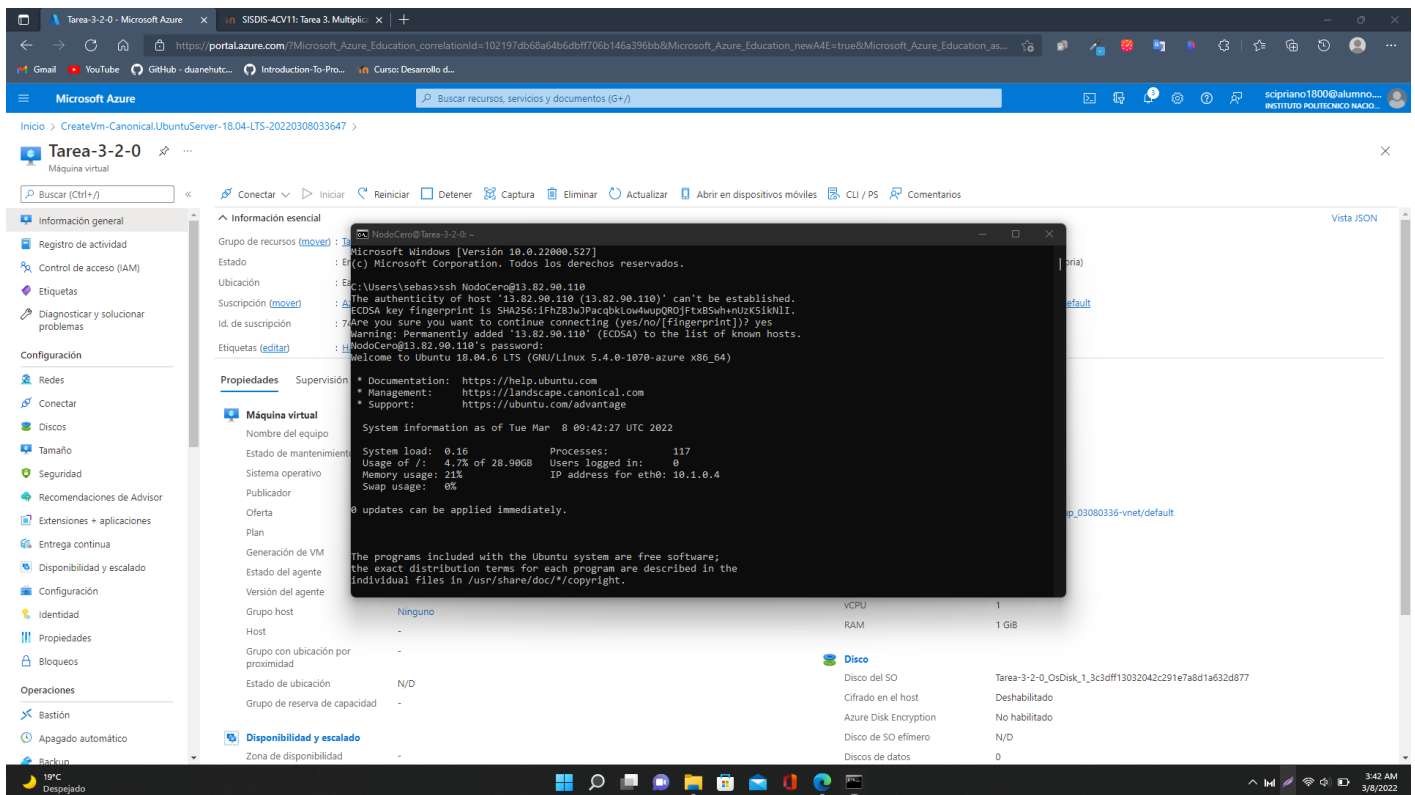


Ilustración 9 Verificación con el uso de la consola.

Ya dentro de la consola, en la imagen de abajo podemos apreciar como ya tenemos acceso a la terminal del modo tal que podemos ingresar los comandos que nosotros queramos sin ningún tipo de problema para a posteriori hacer las pruebas pertinentes.

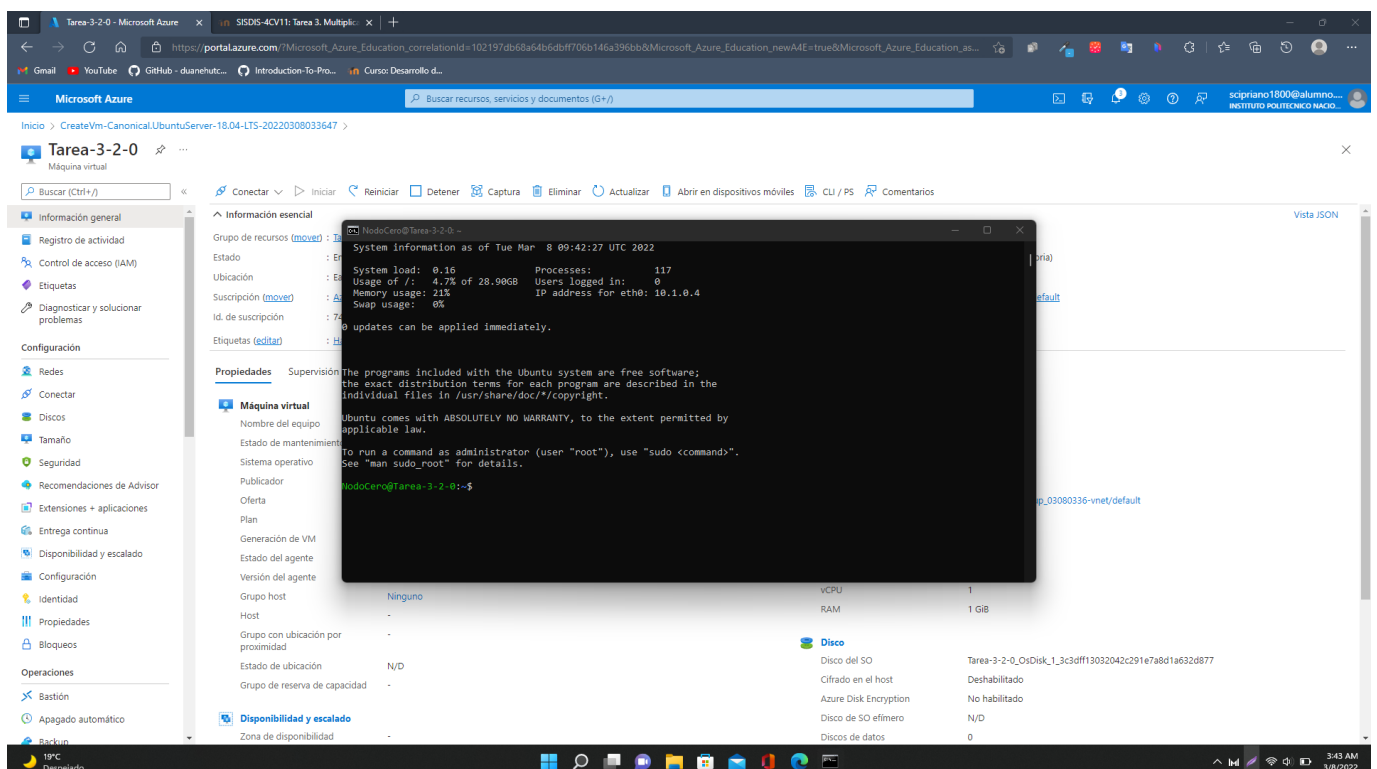


Ilustración 10 Acceso de la terminal.

Debemos hacer un update para poder trabajar con algunas de las herramientas del sistema actualizadas de modo tal que no tengamos ningún tipo de problema al estar trabajando y haciendo las pruebas con nuestro programa. Esto lo podemos ver en la imagen que aparece en la parte de abajo.

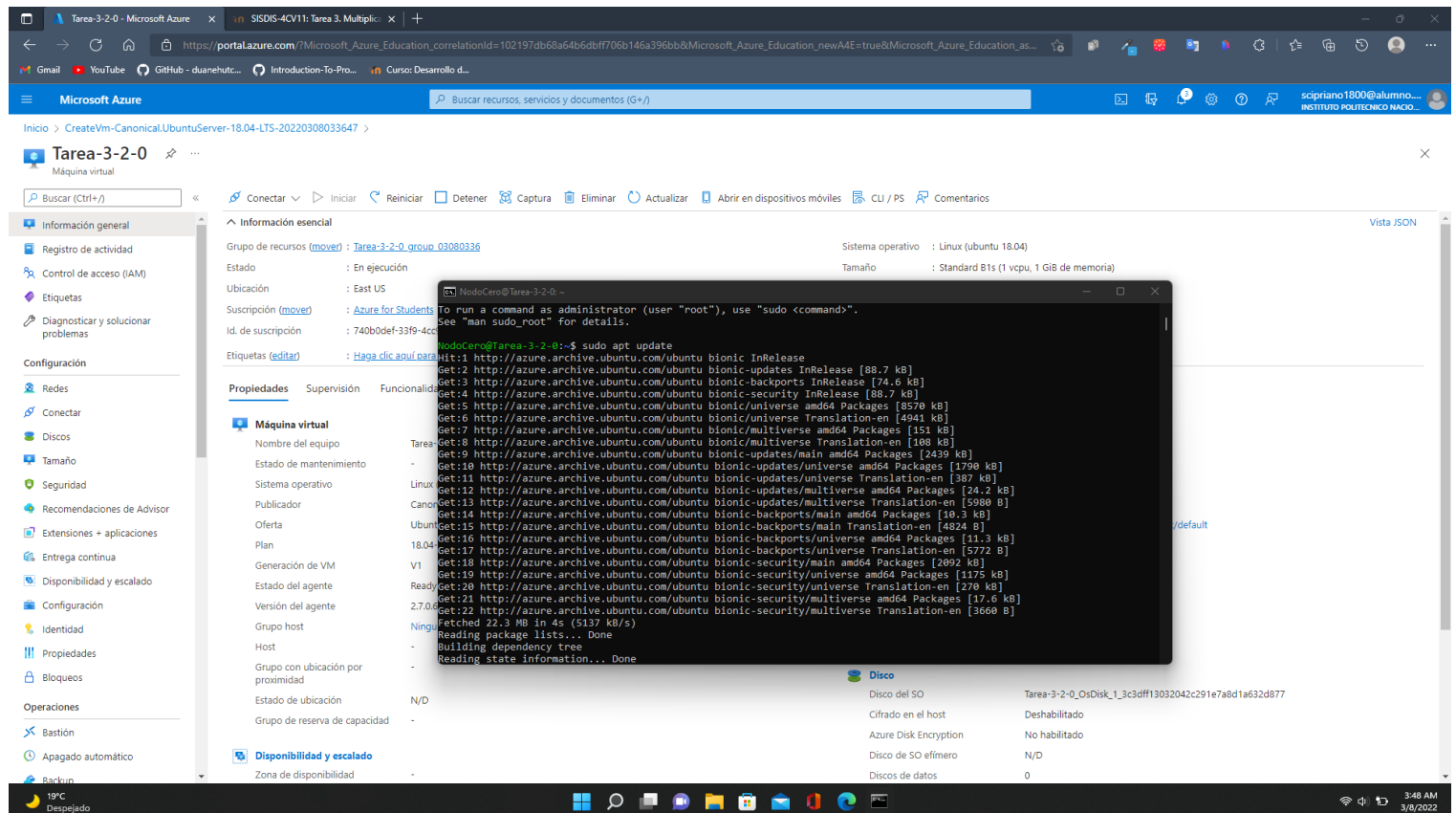


Ilustración 11 Actualizaciones dentro del sistema de la máquina virtual.

Ahora bien, es pertinente hacer la instalación de las herramientas para el trabajo en las pruebas del programa, debemos instalar el jre de modo tal que nosotros podamos trabajar con la última versión y así evitar problemas.

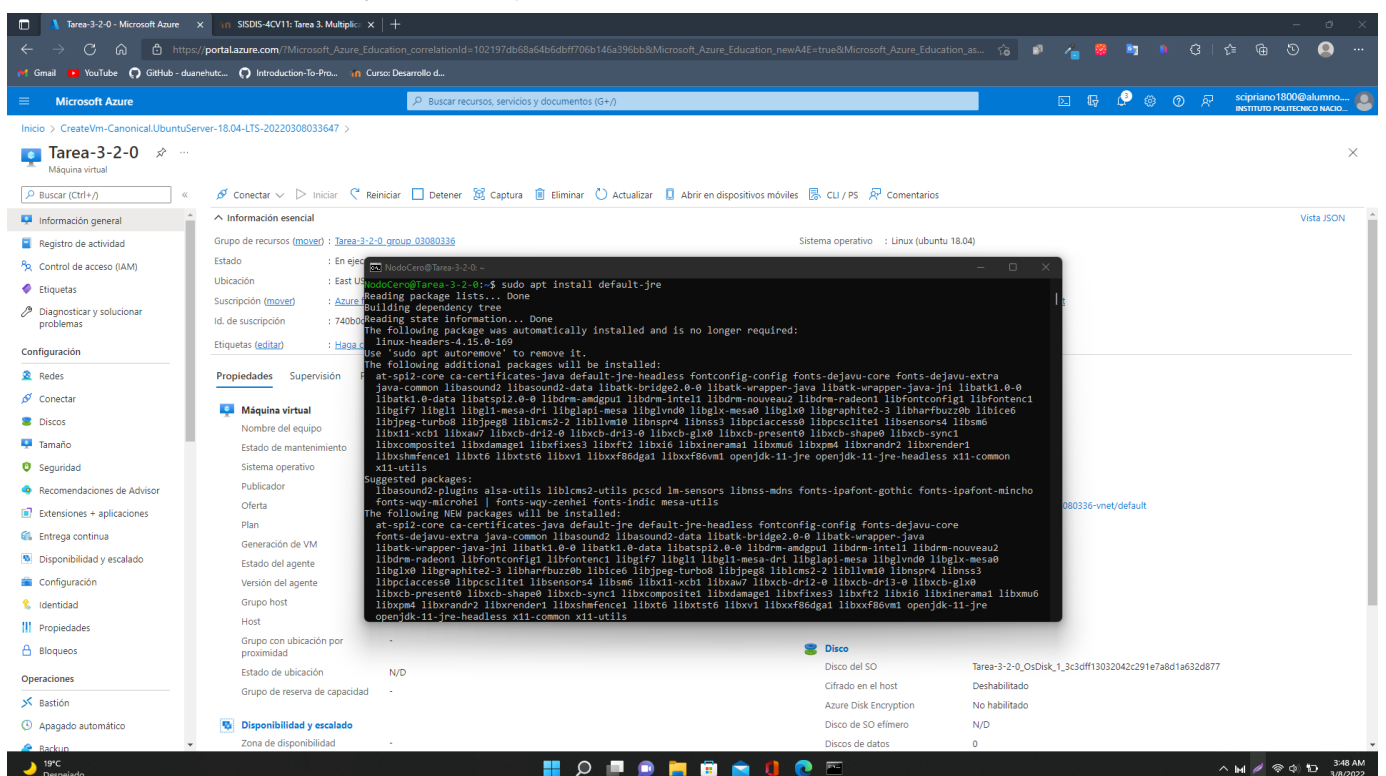


Ilustración 12 Instalación del JRE

Ahora nosotros tenemos que hacer la instalación del JDK, para lo cual lo más optimo en Ubuntu es hacerlo desde la consola y hace como en la imagen anterior obtener la versión más reciente posible y evitar fallos.

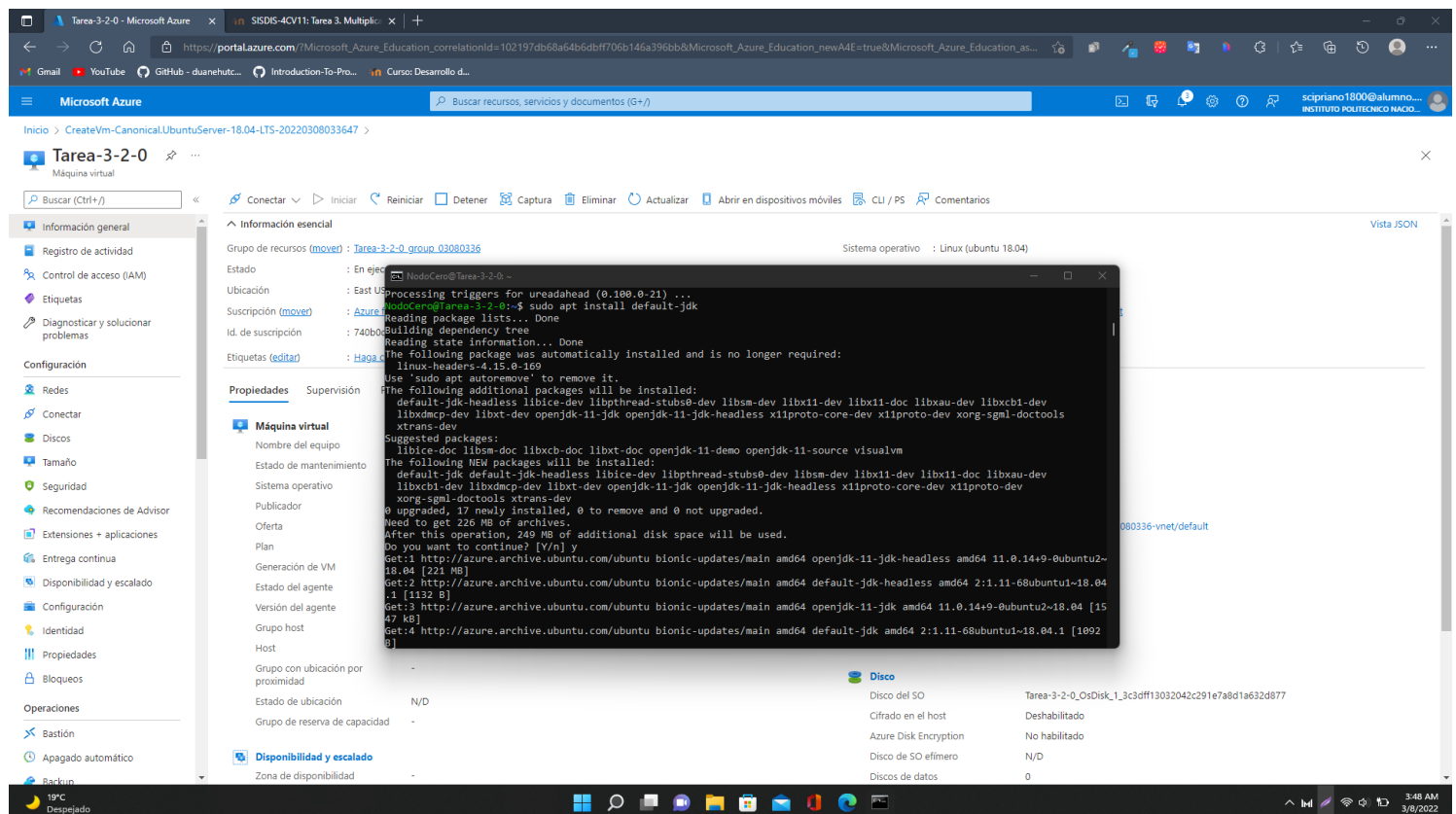


Ilustración 13 Instalación del JDK dentro de la máquina virtual.

Una vez hecho lo anterior debemos ingresar nuestra contraseña y el usuario para poder hacer la conexión, podemos ver como el sistema pudo hacer la conexión de manera optima desde la consola sin ningún tipo de problema.

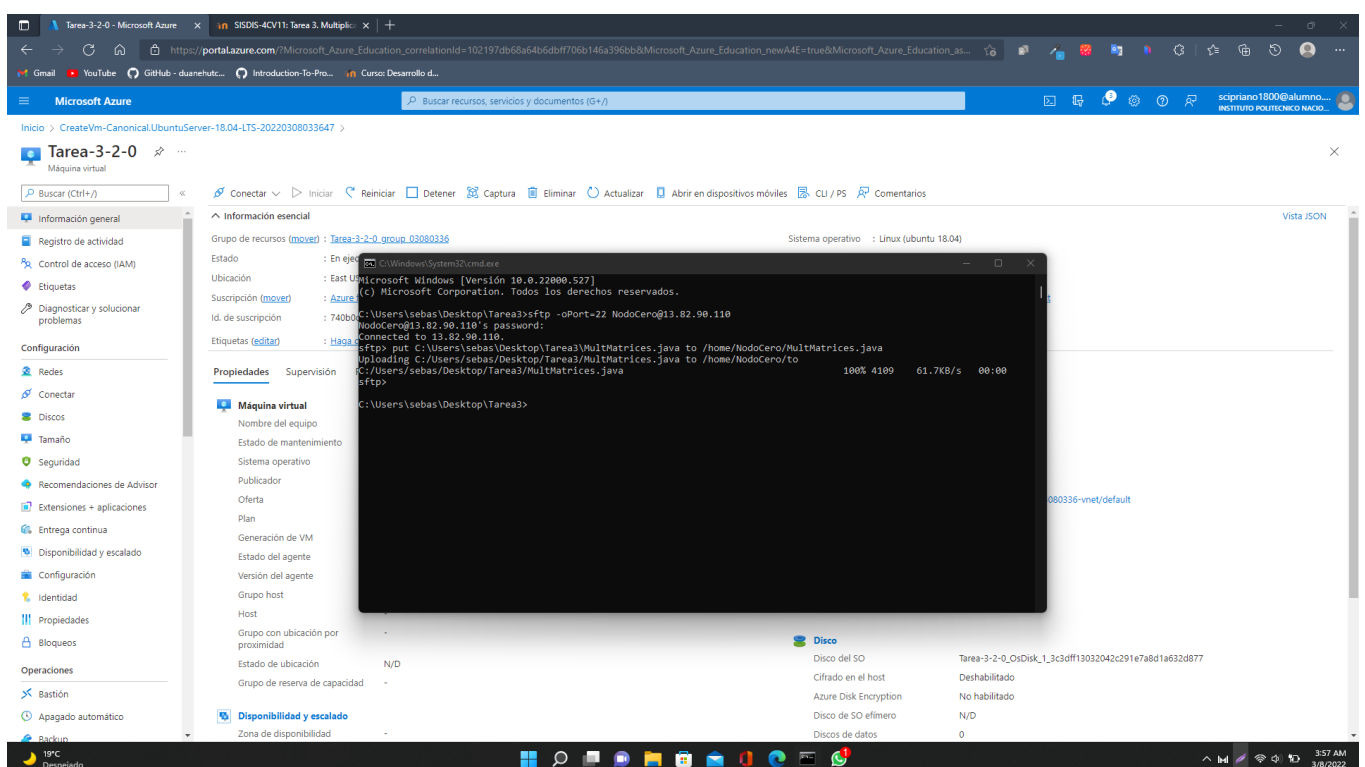


Ilustración 14 Conexión dentro de la consola.

Podemos desde nano, acceder a nuestro código del programa, esto lo podemos ver en la siguiente imagen desde la consola del sistema; esto es muy útil porque podemos trabajar de manera directa con el programa para hacer las pruebas.

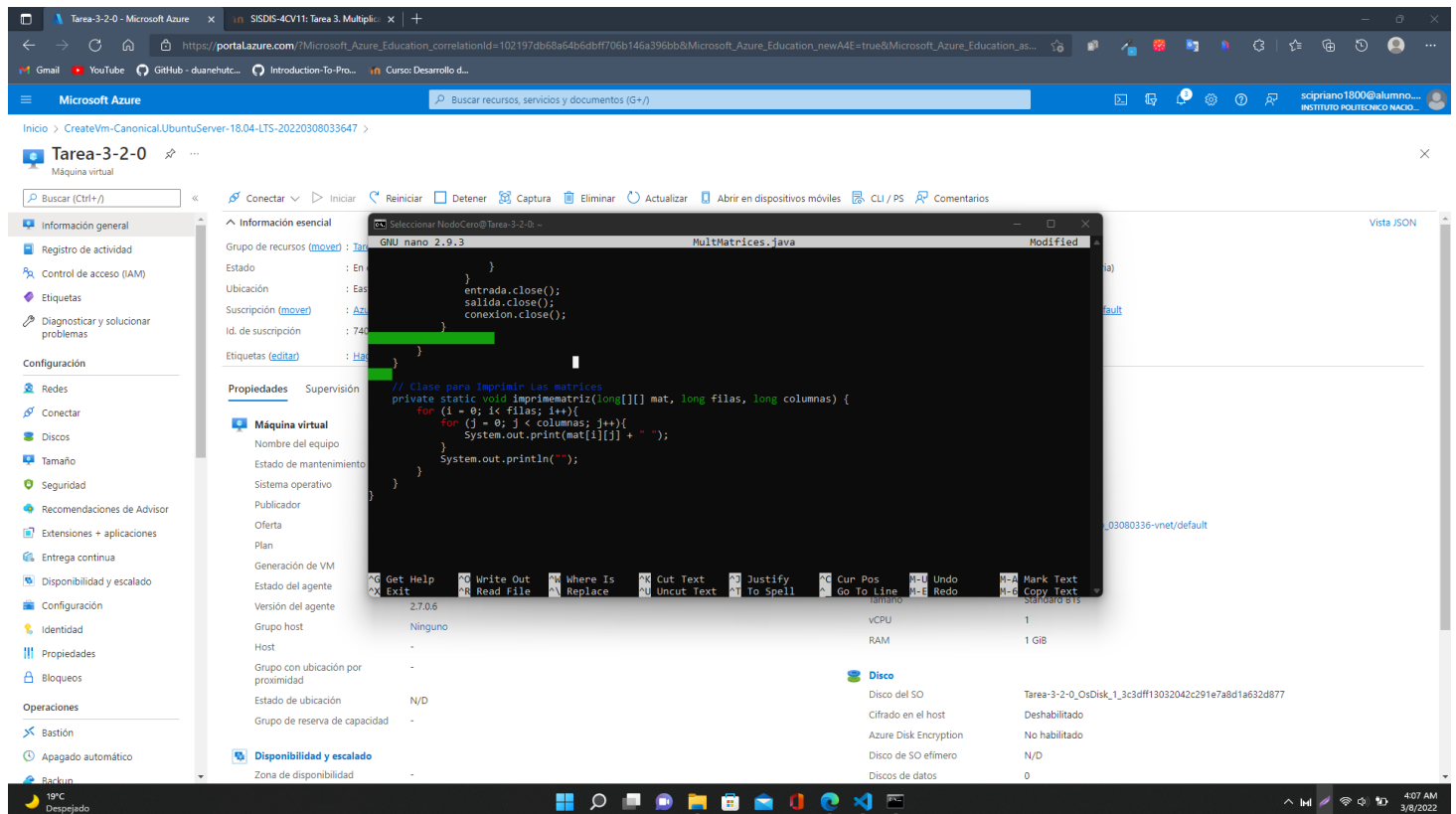


Ilustración 15 Apreciación del código desde el editor de nano.

Si checamos la sección de las redes podemos ver como tenemos las reglas de puerto de entrada, y podemos del mismo modo agregar las reglas de seguridad de entrada que nosotros queramos; esto sin duda es de mucha ayuda para poder personalizar todo a como nosotros queramos, del mismo modo hay que recalcar que esto lo vemos en la imagen de abajo.

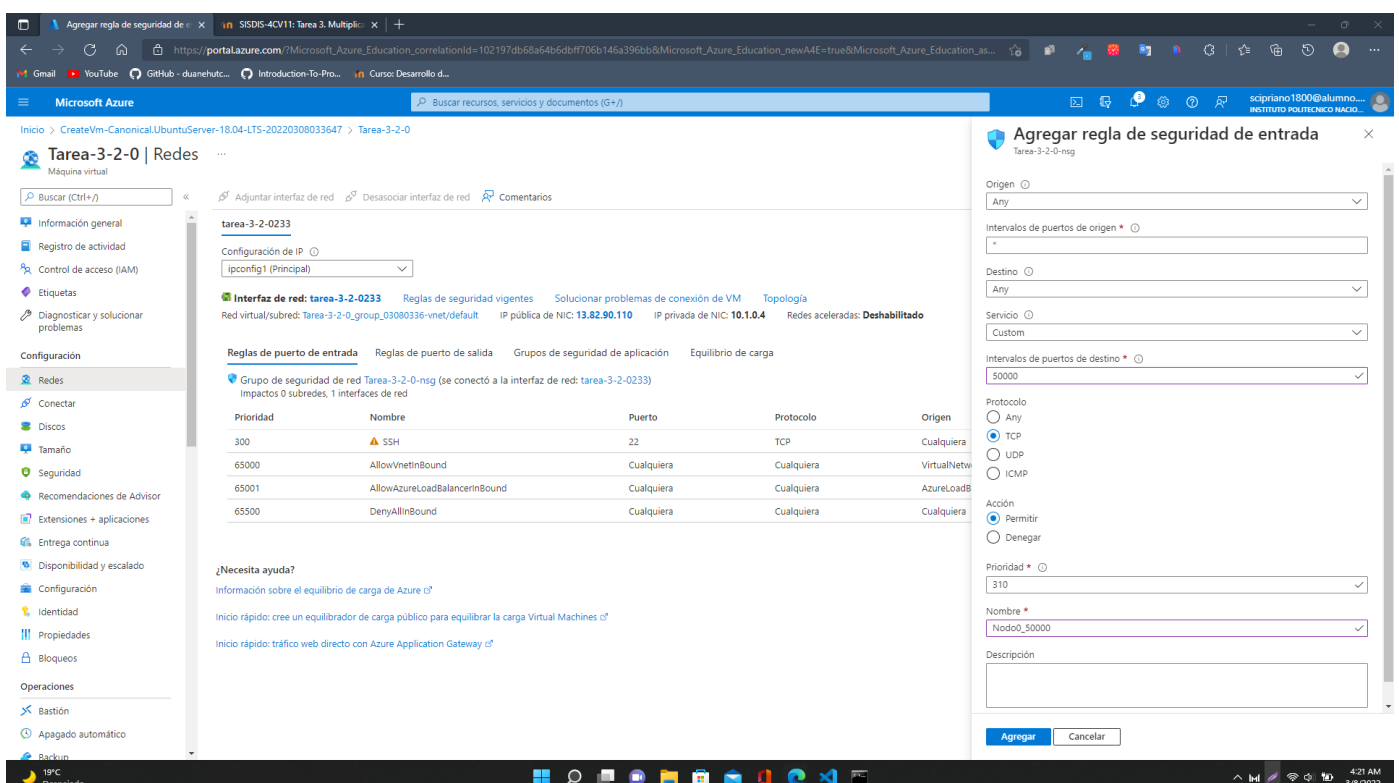


Ilustración 16 Sección de redes de la máquina virtual.

Podemos ver como hemos creado todas nuestras máquinas virtuales con el nombre que usted nos indicó de acuerdo con el número de equipo y de acuerdo con el número de la tarea.

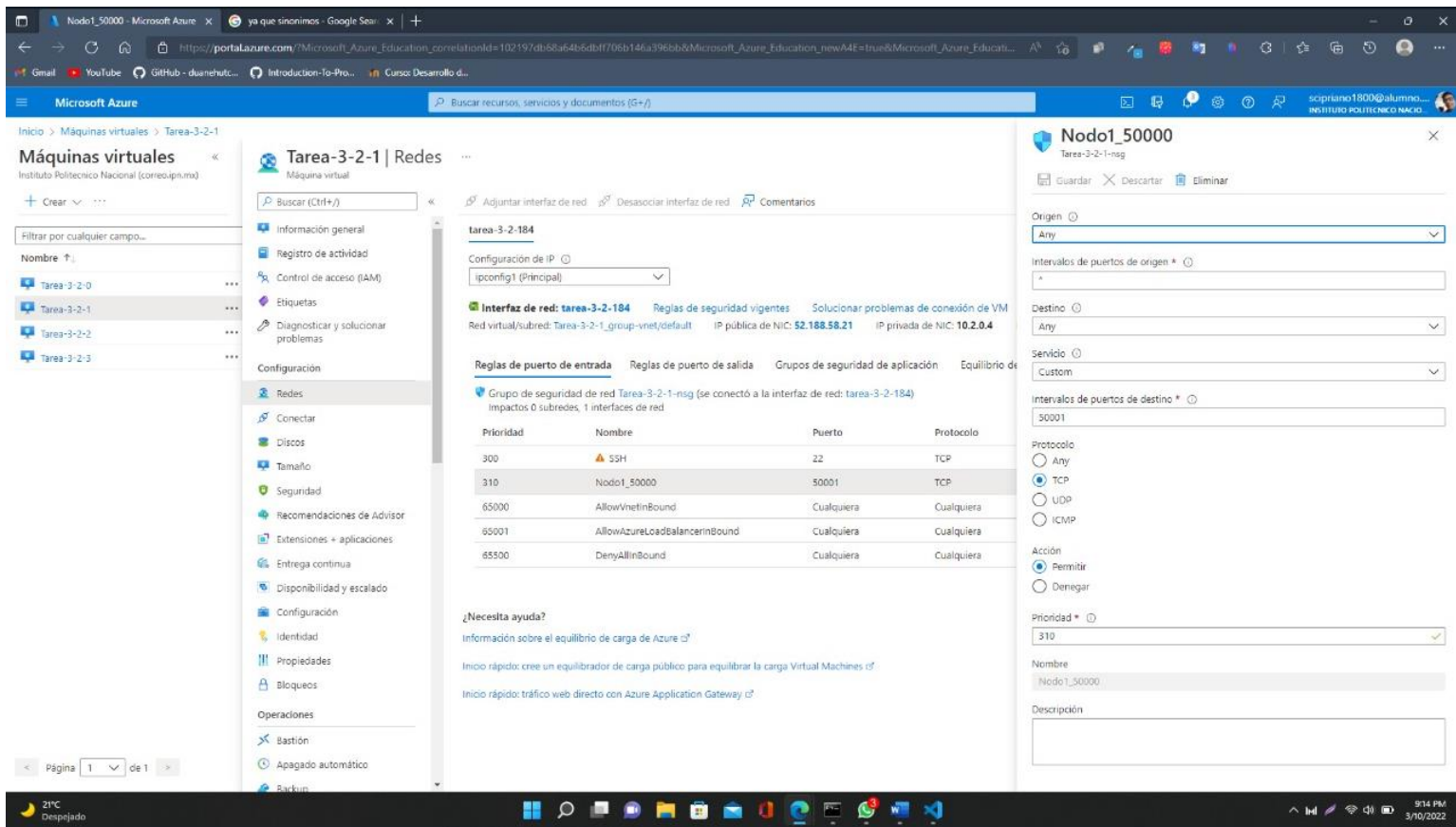


Ilustración 17 Muestra de la creación de todas las máquinas virtuales necesarias.

Vamos a mostrar a continuación como podemos hacer cambios en el archivo, para lo cual abriremos nuestro trabajo con el nano y haremos un pequeño cambio en los comentarios.

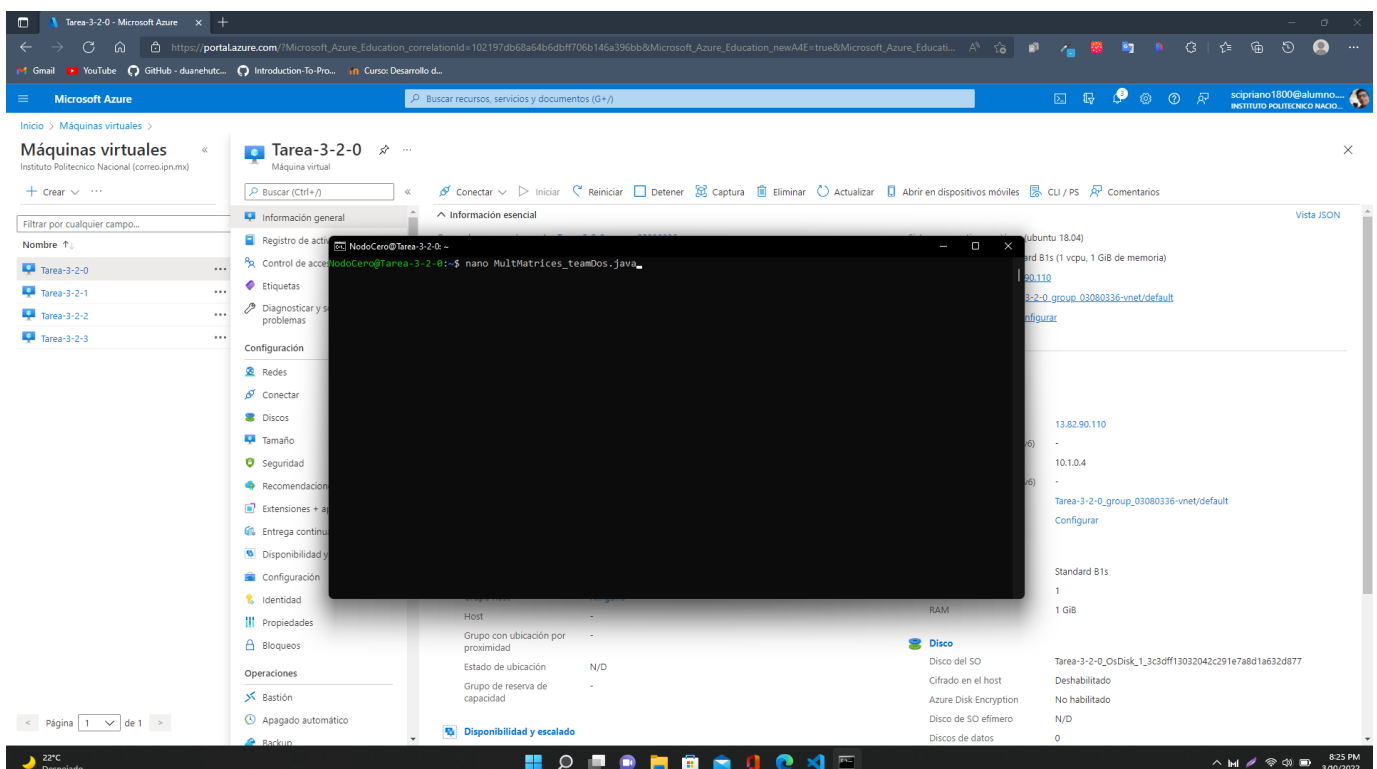


Ilustración 18 Abrimos el archivo en nano para su posterior edición.

Lo que corresponde es ahora hacer la modificación que mencionamos, para distinguir los dos archivos la modificación ha sido hecha en el nombre, ya que a nuestro archivo lo hemos nombrado “MultMatrices_teamDos.java”, hemos cambiado pequeñas cosas en el archivo para verificar que es posible y que funciona de manera óptima.

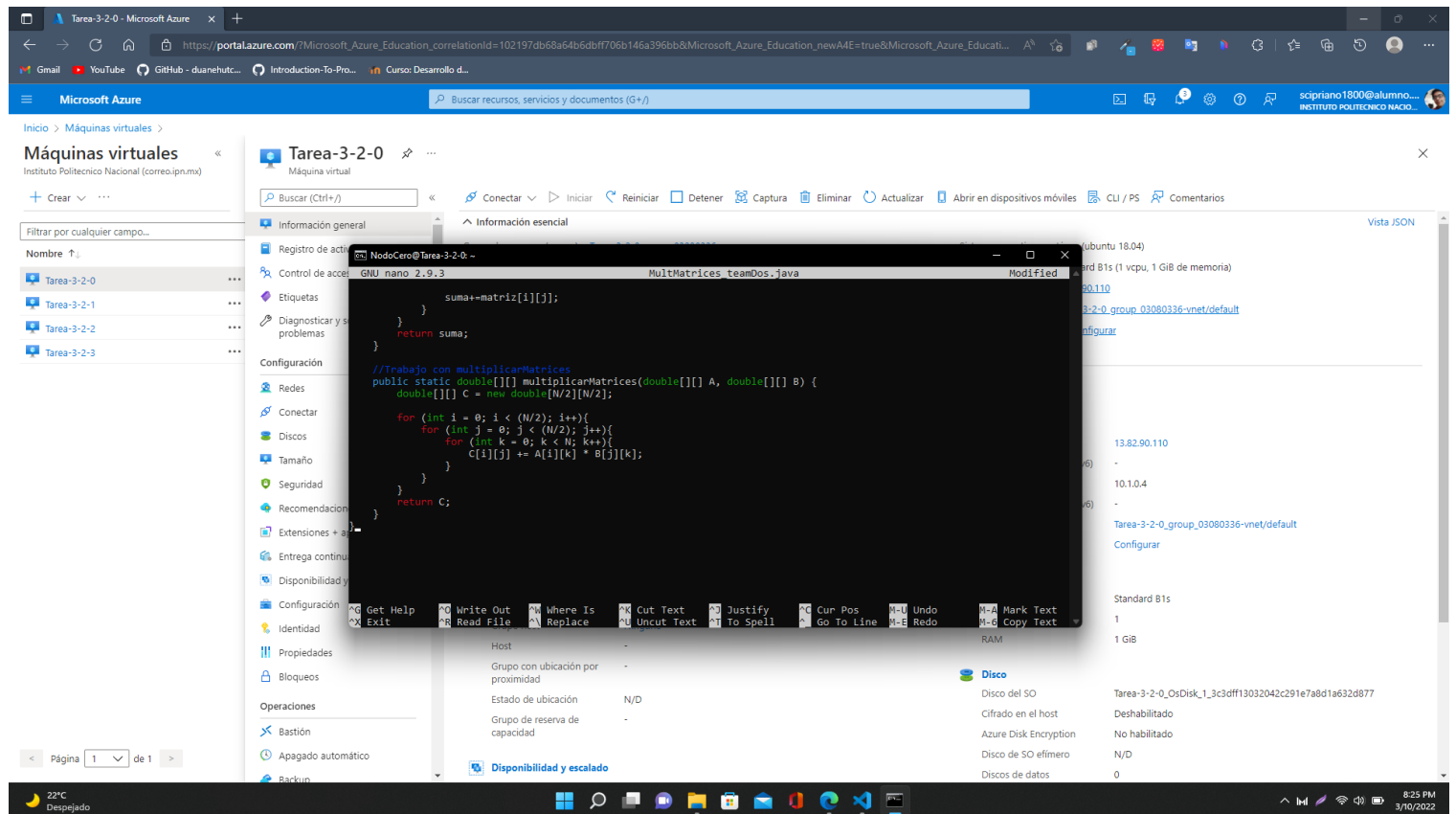


Ilustración 19 Modificación del archivo original para verificar que todo está en funcionamiento.

Vamos a mostrar el trabajo, para lo cual tenemos que abrir todas y cada una de nuestras máquinas virtuales, para lo cual estamos mostrando en la parte superior la VM1 Y VM3 y en la parte inferior la VM2 y la VM0; en estas pantallas se mostraran las operaciones en simultaneo.

Por cuestiones de espacio en la hoja lo hemos puesto en la siguiente hoja del documento, de modo tal que pueda ser más visible. Cabe destacar que las máquinas están nombradas como nos indicó, para que verifique la autenticidad de estas.

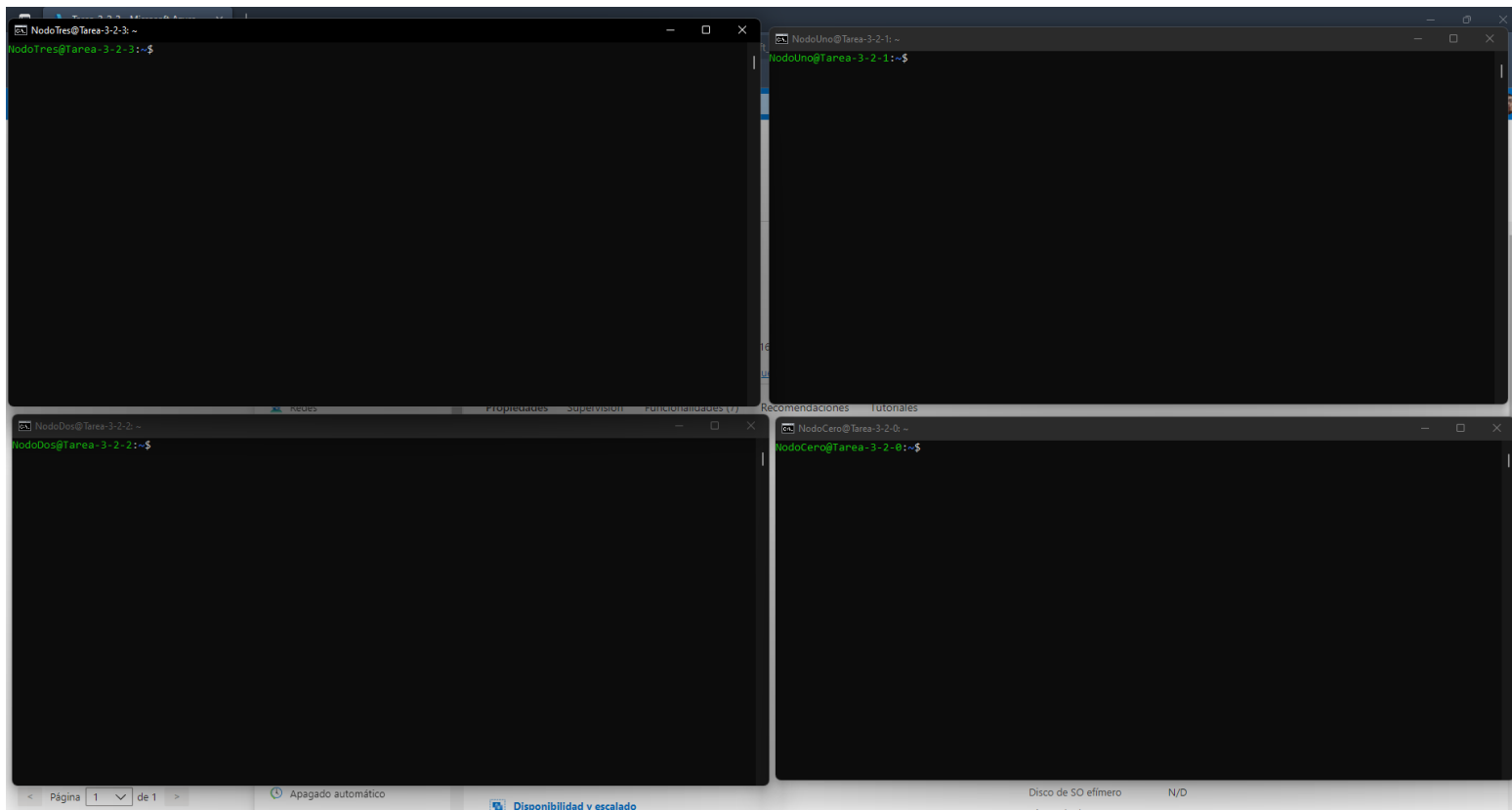


Ilustración 20 Despliegue de todas las consolas de cada una de las máquinas virtuales.

El programa ya ha sido compilado, pero lo volveremos a compilar a posteriori para el caso de $N = 1000$, ahora en la imagen de abajo lo que haremos es la evaluación del caso de $N = 8$, para lo cual primero hemos de correr en nuestros nodos periféricos el programa, en los nodos 1, 2, 3; para lo cual recibiremos un mensaje que nos indicará que todo está listo.

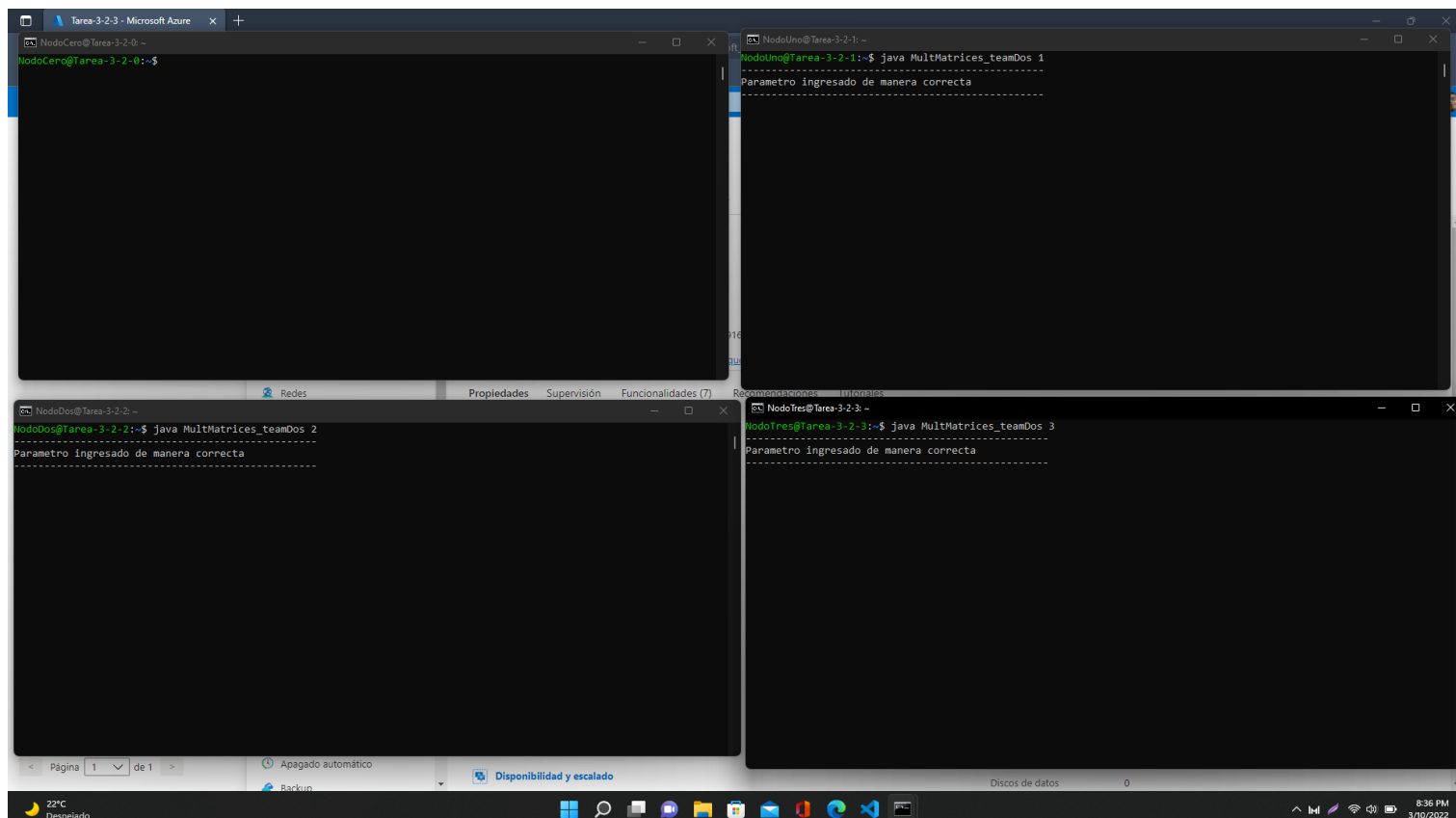


Ilustración 21 Ejecución del programa para los nodos 1,2,3 en el caso de $N = 8$

Podemos ver el despliegue de las matrices en cada una de las consolas de los nodos, y vemos que el valor de nuestro checksum es 217728.0 para el caso de $N = 8$.

```
NodoCero@Tarea-3-2-0:~$ java MultMatrices_teamDos 0
Parametro ingresado de manera correcta
-----
Matriz C:
-----
3500.0 3360.0 3220.0 3080.0 2940.0 2800.0 2660.0 2520.0
3640.0 3492.0 3344.0 3196.0 3048.0 2900.0 2752.0 2604.0
3780.0 3624.0 3468.0 3312.0 3156.0 3000.0 2844.0 2688.0
3920.0 3756.0 3592.0 3428.0 3264.0 3100.0 2936.0 2772.0
4060.0 3888.0 3716.0 3544.0 3372.0 3200.0 3028.0 2856.0
4200.0 4020.0 3840.0 3660.0 3480.0 3300.0 3120.0 2940.0
4340.0 4152.0 3964.0 3776.0 3588.0 3400.0 3212.0 3024.0
4480.0 4284.0 4088.0 3892.0 3696.0 3500.0 3304.0 3108.0
-----
Checksum total ==> 217728.0
-----
NodoCero@Tarea-3-2-0:~$

NodoUno@Tarea-3-2-1:~$
3.0 8.0 13.0 18.0 23.0 28.0 33.0 38.0
-----
Matriz B
-----
0.0 5.0 10.0 15.0 20.0 25.0 30.0 35.0
-1.0 4.0 9.0 14.0 19.0 24.0 29.0 34.0
-2.0 3.0 8.0 13.0 18.0 23.0 28.0 33.0
-3.0 2.0 7.0 12.0 17.0 22.0 27.0 32.0
-----
Matriz C
-----
3500.0 3360.0 3220.0 3080.0
3640.0 3492.0 3344.0 3196.0
3780.0 3624.0 3468.0 3312.0
3920.0 3756.0 3592.0 3428.0
-----
NodoUno@Tarea-3-2-1:~$

NodoDos@Tarea-3-2-2:~$
3.0 8.0 13.0 18.0 23.0 28.0 33.0 38.0
-----
Matriz B
-----
-4.0 1.0 6.0 11.0 16.0 21.0 26.0 31.0
-5.0 0.0 5.0 10.0 15.0 20.0 25.0 30.0
-6.0 -1.0 4.0 9.0 14.0 19.0 24.0 29.0
-7.0 -2.0 3.0 8.0 13.0 18.0 23.0 28.0
-----
Matriz C
-----
2940.0 2800.0 2660.0 2520.0
3048.0 2900.0 2752.0 2604.0
3156.0 3000.0 2844.0 2688.0
3264.0 3100.0 2936.0 2772.0
-----
NodoDos@Tarea-3-2-2:~$

NodoTres@Tarea-3-2-3:~$
7.0 12.0 17.0 22.0 27.0 32.0 37.0 42.0
-----
Matriz B
-----
0.0 5.0 10.0 15.0 20.0 25.0 30.0 35.0
-1.0 4.0 9.0 14.0 19.0 24.0 29.0 34.0
-2.0 3.0 8.0 13.0 18.0 23.0 28.0 33.0
-3.0 2.0 7.0 12.0 17.0 22.0 27.0 32.0
-----
Matriz C
-----
4060.0 3888.0 3716.0 3544.0
4200.0 4020.0 3840.0 3660.0
4340.0 4152.0 3964.0 3776.0
4480.0 4284.0 4088.0 3892.0
-----
NodoTres@Tarea-3-2-3:~$
```

Ilustración 22 Ejecución final del código para el caso de $N = 8$

Ahora, necesitamos volver a compilar nuestro código, lo cual podemos verlo en la consola de la VM 1 y ya luego tenemos la ejecución del programa en las distintas consolas, para lo cual recibiremos exactamente el mensaje de que todo está listo para proceder. Recordemos que esta captura pertenece al caso de $N = 1000$.

```
NodoCero@Tarea-3-2-0:~$
NodoCero@Tarea-3-2-0:~$

NodoUno@Tarea-3-2-1:~$
NodoUno@Tarea-3-2-1:~$ java MultMatrices_teamDos.java
Parametro ingresado de manera correcta
-----

NodoDos@Tarea-3-2-2:~$
NodoDos@Tarea-3-2-2:~$ javac MultMatrices_teamDos.java
NodoDos@Tarea-3-2-2:~$ java MultMatrices_teamDos 2
Parametro ingresado de manera correcta
-----

NodoTres@Tarea-3-2-3:~$
NodoTres@Tarea-3-2-3:~$ javac MultMatrices_teamDos.java
NodoTres@Tarea-3-2-3:~$ java MultMatrices_teamDos 3
Parametro ingresado de manera correcta
-----
```

Ilustración 23 Ejecución del programa en el caso de $N = 1000$

Vemos que en la siguiente imagen nos quedan todas nuestras matrices, pero del mismo modo tenemos nuestro checksum para para este caso es de 8.07133725E15, (esto lo podemos apreciar en la consola superior del NodoCero)

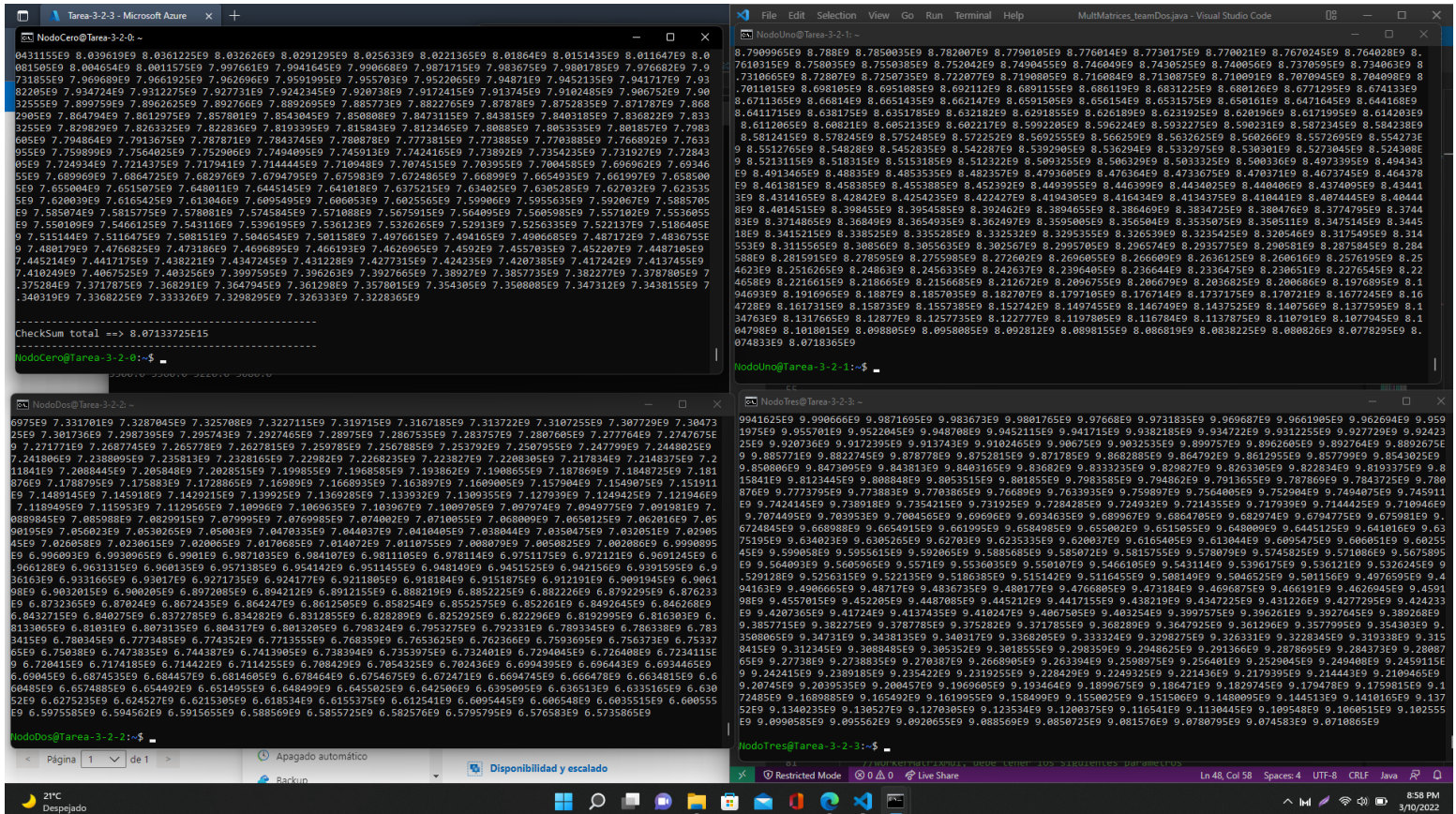


Ilustración 24 Ejecución final del programa para el caso de N = 1000

Especificaciones sobre el código y proceso de trabajo.

Durante la creación de nuestro código, al inicio estuvimos trabajando de modo tal que todas las operaciones se hacían de manera tal que la consola uno ejecutaba el código y a posteriori seguía con las siguientes operaciones en la consola, pero más sin en cambio lo que hicimos después fue corregir este error para que el código corriera todas las operaciones de manera simultánea.

Ahora bien, una cosa que también hemos considerado es tratar de optimizar nuestro sistema y poder trabajar con el uso de los hilos de modo tal que los procesos puedan apresurarse y sea mucho más sencillo su manejo.

Estos aspectos fueron muy importantes para nosotros, ya que consideramos que es necesario el trabajo del programa de manera simultánea de modo tal que podamos aprovechar los principios del sistema distribuido y sobre todo logremos hacer uso de la estructura que se nos ha propuesto en la práctica de este curso.

Conclusiones

Cazares Martínez Maximiliano

Esta práctica nos mostró lo interesante y poderoso que son los sistemas distribuidos y como el cálculo distribuido puede hacer que un proceso pesado para un solo equipo se convierta en uno más ligero y realizado por varios equipos. Esta práctica fue un poco complicada debido a que se trató el rol de cliente y servidor de una forma distinta a lo que habíamos estado trabajando, pero al final logramos terminarla con éxito.

Chavarría Vázquez Luis Enrique

Esta ocasión la práctica si que fue un poco más desafiante por la cantidad de especificaciones y requerimientos que se tienen que cumplir para satisfacer las necesidades y los resultados requeridos, si bien fue un problema que hemos podido manejar una de las cosas que más me costaron en lo personal retomar fue la parte de entender el problema puesto que me cuesta un poco de trabajo ver problema donde se usan temáticas relacionadas con las matemáticas para ejemplificar algo que aún estoy aprendiendo, puesto que en ocasiones esto aumenta la abstracción del problema, pero vasto con revisar apuntes y las notas de la clase para poder volver a retomar los aspectos esenciales y poner manos a la obra.

Una de las dificultades que tuvimos en el equipo fue poder hacer que las operaciones se realizaran de manera simultánea, por lo que teníamos en principio la duda de donde hacer que diera inicio nuestras operaciones, por lo que decidimos desde el nodo 0 hacer que iniciase la instrucción de modo tal que no quedará aislado el código en el mismo.

También hay que destacar que la facilidad del uso de las interfaces en el Azure es bastante buena, porque en realidad el servicio esta muy bien documentado y se nos iba guiando en cada una de las parte del proceso; esto sin duda es un gran apoyo de cara a poder hacer algunas aplicaciones mucho más complejas en el futuro y poder hacer nuevos experimentos, aunque por otra parte tengo que decir que hasta el momento Azure no se me ha hecho tan flexible como AWS y como Heroku, pero quiero pensar que es cuestión de la falta de costumbre y de como es que suelo trabajar.

Finalmente quiero decir que me emociona poder ver aplicaciones que tengan un uso más convencional en el futuro y poder sobre todo hacer experimentos enfocados a la experiencia del usuario y sobre todo a la eficiencia de los sistemas para que determinadas operaciones corran de manera adecuada.

Cipriano Damián Sebastián.

De acuerdo con lo aprendido en esta práctica, se puede llegar a la conclusión de que es muy importante para los Ingenieros en Sistemas Computacionales conocer y saber utilizar los recursos proporcionados por el computo en la nube, pues estas tecnologías facilitan la implementación, el uso y mantenimiento de los sistemas. Además, Resulta interesante aplicar los conocimientos adquiridos en el curso, pero con máquinas virtuales en la nube, ya que es posible ver de una forma más realista el funcionamiento de los sistemas distribuidos.