

Instituto Politécnico Nacional.  
Escuela Superior de Cómputo

(ESCOM).

Desarrollo de Sistemas Distribuidos.

Tarea 6.  
“Multiplicación de matrices utilizando objetos distribuidos”.

Profesor Carlos Pineda   
Alumno: Sánchez Martínez Eli   
Grupo: 4CM5

20/11/2020

Objetivo: Cada alumno deberá desarrollar un sistema que calcule el producto de dos matrices cuadradas utilizando Java RMI, tal como se explicó en clase.

**Desarrolló:**

Al igual que la tarea 3, se hará la multiplicación de matrices distribuidas, pero esta vez se ejecutará usando RMI, primeramente, se deberá implementar la siguiente topología de red.

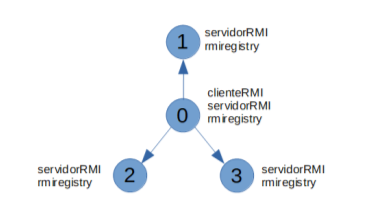
Se deberá ejecutar dos casos:

* N=4, se deberá desplegar las matrices A, B y C y el checksum de la matriz C.
* N=500, deberá desplegar el checksum de la matriz C.

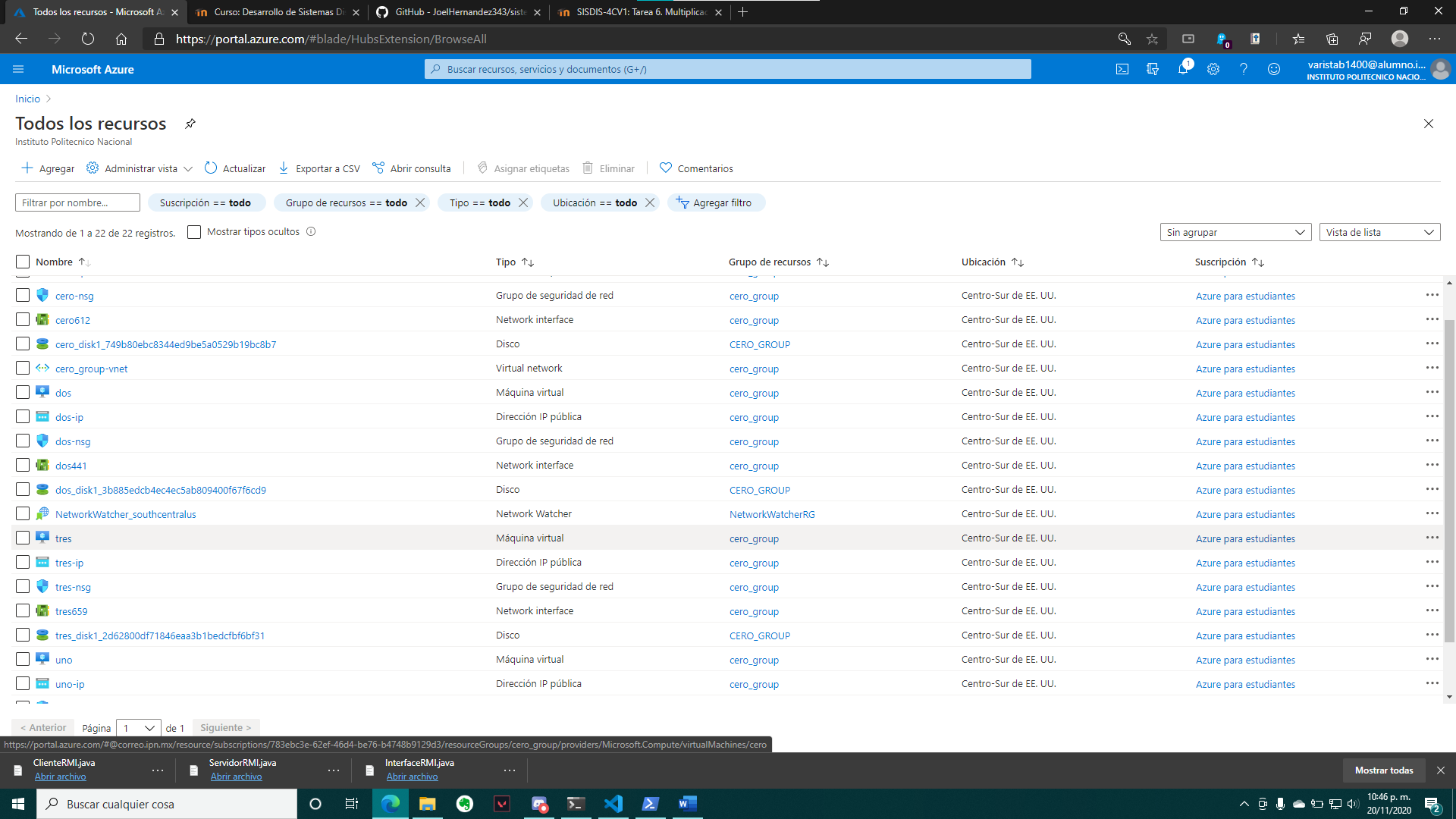
Los elementos de las matrices serán de tipo int y el checksum será de tipo long.

Se deberá inicializar las matrices A y B de la siguiente manera (notar que la inicialización es diferente a la que se realizó en la tarea 3):

* A[i][j]=2 \* i - j
* B[i][j] = 2 \* i + j

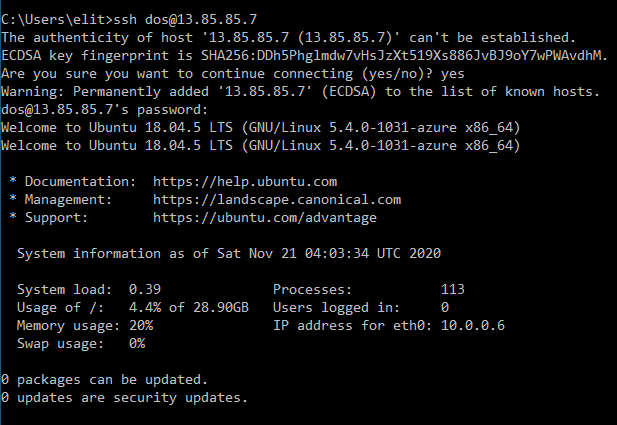


Para poder realizar la práctica es importante destacar que las máquinas virtuales deben crearse en el mismo grupo de recursos, a modo que se pueda utilizar su dirección IP privada para comunicarse entre ellas.



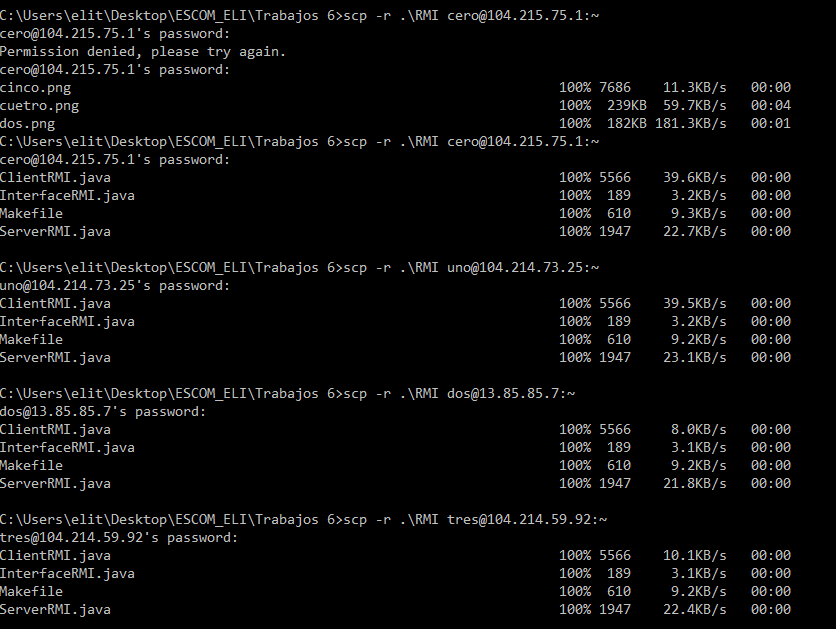
**1.**Recursos creados para cada VM.

Una vez que tenemos las máquinas listas para usarse, se procede a explicarse el proceso que se hizo en cada máquina virtual con una máquina de ejemplo.



**2.** Ingresando a cada máquina virtual.

Procedemos a cargar los archivos a cada máquina virtual:

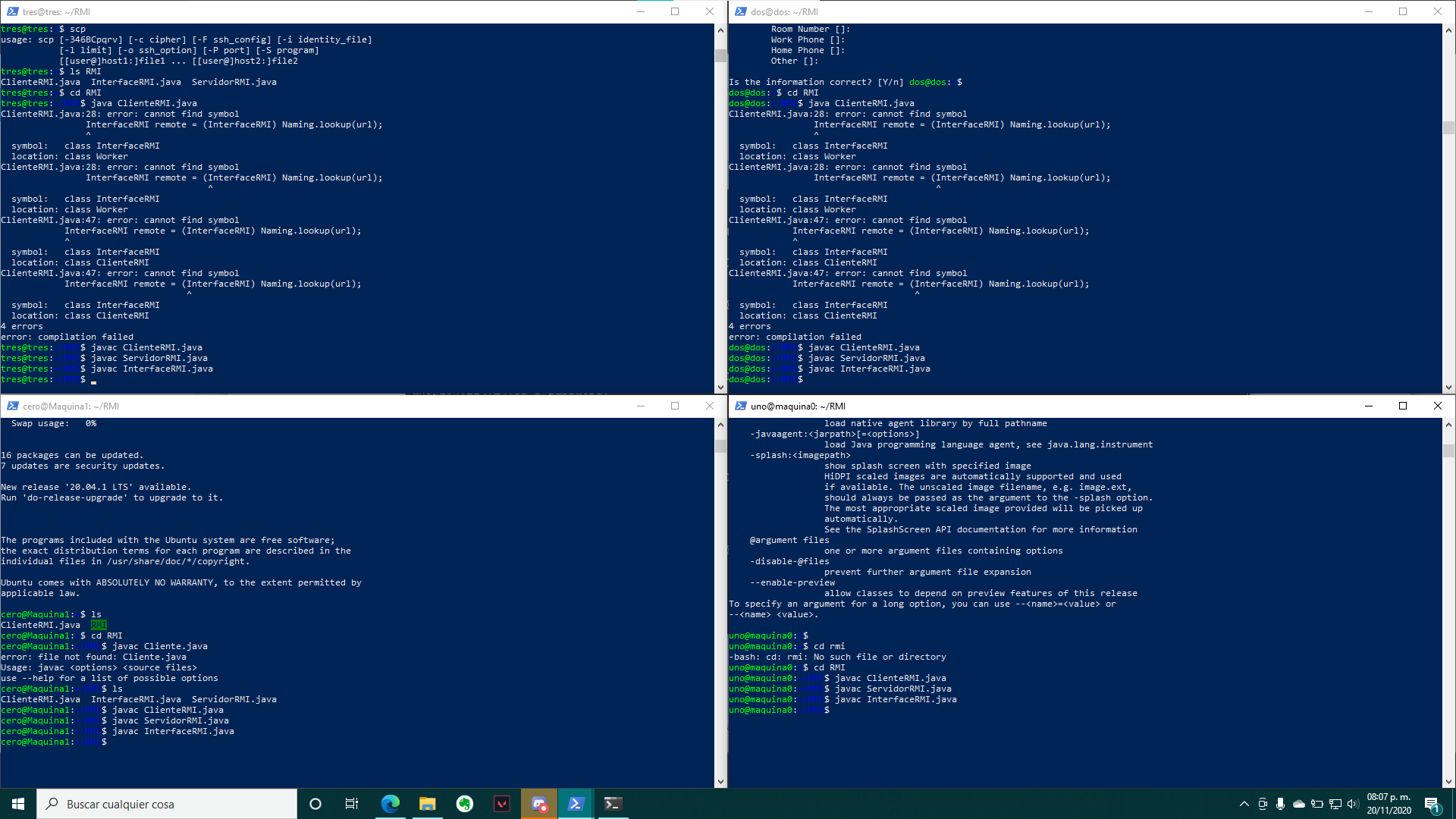


**3.** Cargando los archivos a cada máquina virtual.

Procedemos a compilar los archivos en cada VM y ejecutamos los comandos, primero compilamos los archivos con terminación java, con el comando:

* javac nombre\_archivo.java
* java nombre\_archivo.

Una vez compilados procedemos a ejecutar cada archivo con su respectivo nodo para poder ver la implementacion de nuestra multiplicación.

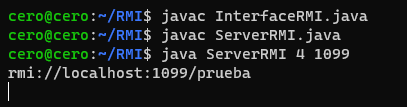


**4**.Compilando todos los archivos con terminación java en las VM.

Para ejecutar tenemos que poner en cada nodo nuestros ejecutables de java con terminación:

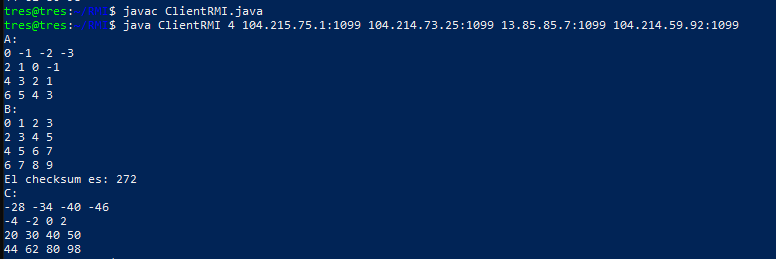
* InterfaceRMI.java
* ServerRMI.java

Y para finalizar ejecutamos nuestro Server con el numero de N y el puerto en este caso, este proceso se repite en 3 nodos.



**5**. Compilando los archivos en un nodo de la VM.

En el cuarto nodo ejecutamos el ClienteRMI.java, dándole nuestro valor de N y también la dirección IP junto con su respectivo puerto que en este caso es de 1099 y observamos el resultado de la multiplicación de nuestra matriz.



**6.** Resultado del cliente de una VM.

Ejecutamos el siguiente caso, en el cual N vale 500, entonces se repite el mismo proceso que con el valor de N 4, solo que se cambia por 500.



**7.** Ejecutando el serverRMI en una VM.

Al hacer eso, solo se tendrá que imprimir el checksum, el cual podemos ver a continuación:



**8**.Resultado del cliente en la VM.

Conclusión.

En esta práctica se puede notar a la perfección la diferencia entre usar objetos remotos, ya que esta misma actividad se realizó en la tarea 3, usar sockets. Esta hizo que fuera una tarea más complicada ya que él envió de los datos lo hacían una tarea complicada.

El usar objetos remotos nos garantiza una forma más sencilla de realizar la tarea además de que la integridad de los datos se garantiza de una mejor manera en contraste a enviar datos por sockets, incluso esto se ve reflejado en la cantidad de líneas de código que se usaron para el programa.