



# Instituto Politécnico Nacional



## Escuela Superior de Cómputo

Chat multicast

TAREA 4

Materia:

Desarrollo De Sistemas Distribuidos

Grupo:

4CV14

Profesor:

Pineda Guerrero Carlos

Alumno:

Castro Cruces Jorge Eduardo

Boleta:

2015080213

Fecha:

Viernes, 24 de septiembre de 2021

## 1. Desarrollo del programa

Desarrollar un solo programa en Java en modo consola que implemente un chat utilizando comunicación multicast mediante datagramas.

Se deberá ejecutar el programa en una máquina virtual con Windows Server 2012 en Azure.

Se deberá pasar como parámetro al programa el nombre del usuario que va escribir en el chat.

Para demostrar el programa se deberá utilizar los siguientes usuarios: hugo, paco y luis (no usar otros usuarios).

El programa deberá utilizar la siguiente función para enviar los mensajes multicast:

```
static void envia_mensaje_multicast(byte[] buffer,String ip,int puerto) throws  
IOException{  
  
    DatagramSocket socket = new DatagramSocket();  
  
    socket.send(new  
DatagramPacket(buffer,buffer.Length,InetAddress.getByName(ip),puerto));  
  
    socket.close();  
}
```

*El programa deberá utilizar la siguiente función para recibir los mensajes multicast:*

```
static      byte[]      recibe_mensaje_multicast(MulticastSocket      socket,int  
Longitud_mensaje) throws IOException  
  
{  
  
    byte[] buffer = new byte[Longitud_mensaje];  
  
    DatagramPacket paquete = new DatagramPacket(buffer,buffer.Length);  
  
    socket.receive(paquete);  
  
    return paquete.getData();  
}
```

El funcionamiento del programa será el siguiente:

El programa creará un thread que actuará como cliente multicast, el cual recibirá los mensajes del resto de los nodos. Cada mensaje recibido será desplegado en la pantalla. El thread desplegará el mensaje que envía el mismo nodo.

En el método main(), dentro de un ciclo infinito:

Se desplegará el siguiente prompt: "Ingrese el mensaje a enviar: " (sin las comillas), entonces se leerá una string (el mensaje).

Se deberá enviar el mensaje a los nodos que pertenecen al grupo identificado por la IP 230.0.0.0 a través del puerto 40000. El paquete a enviar deberá tener la siguiente forma: nombre\_usuario:mensaje\_ingresado dónde nombre\_usuario es el nombre del usuario que pasó como parámetro al programa (hugo, paco o luis) y mensaje\_ingresado el mensaje que el usuario ingresó por el teclado, separados por un el caracter ":" sin espacios.

Se deberá completar el siguiente programa:

```
class Chat

{

    static class Worker extends Thread

    {

        public void run()

        {

            // En un ciclo infinito se recibirán los mensajes enviados al

            // grupo 230.0.0.0 a través del puerto 40000 y se desplegarán en la pantalla.

        }

    }

    public static void main(String[] args) throws Exception

    {

        new Worker().start();

        String nombre = args[0];

        // En un ciclo infinito se leerá cada mensaje del teclado y se enviará el
mensaje al

        // grupo 230.0.0.0 a través del puerto 40000.

    }

}
```

Para probar el programa, se deberá ejecutar la siguiente conversación (la cual aparece en negritas) en tres ventanas de comandos (cmd) en la máquina virtual con Windows 10. En la primera ventana escribirá hugo, en la segunda ventana escribirá paco y en la tercera ventana escribirá luis:

*hugo debe escribir:*

*hola a todos*

*paco debe escribir:*

*hola hugo*

*luis debe escribir:*

*hola hugo*

*hugo debe escribir:*

*¿saben dónde será la fiesta?*

*paco debe escribir:*

*será en mi casa*

*hugo debe escribir:*

*¿a qué hora?*

*luis debe escribir:*

*a las 7 PM*

*hugo debe escribir:*

*adios*

*paco debe escribir:*

*adios hugo*

*luis debe escribir:*

*adios hugo*

Los signos de interrogación y las letras acentuadas deberán desplegarse correctamente en la ventana de comandos de Windows (sugerencia, utilizar la codificación de texto que corresponda).

Se deberá subir a la plataforma: 1) un archivo texto con el código fuente del programa desarrollado y 2) un reporte de la tarea en formato PDF con portada, desarrollo y conclusiones como mínimo. El reporte deberá incluir las capturas de pantalla de la compilación y ejecución del programa, se deberá incluir la captura de pantalla correspondiente a cada paso del procedimiento de creación de la máquina virtual, tal como lo hicimos en clase.

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

El nombre de la máquina virtual deberá ser la letra W y el número de boleta del alumno, si el número de boleta del alumno es 12345678, entonces la máquina virtual deberá llamarse: W12345678. No se admitirá la tarea si la máquina virtual no se nombra como se indicó anteriormente.

El programa a desarrollar deberá ser en modo consola, no se admitirá un programa gráfico.

Recuerden que se debe eliminar la máquina virtual cuando ya no se use, con la finalidad de ahorrar el saldo de sus cuentas de Azure.

Valor de la tarea: 30% (2.1 puntos de la segunda evaluación parcial)

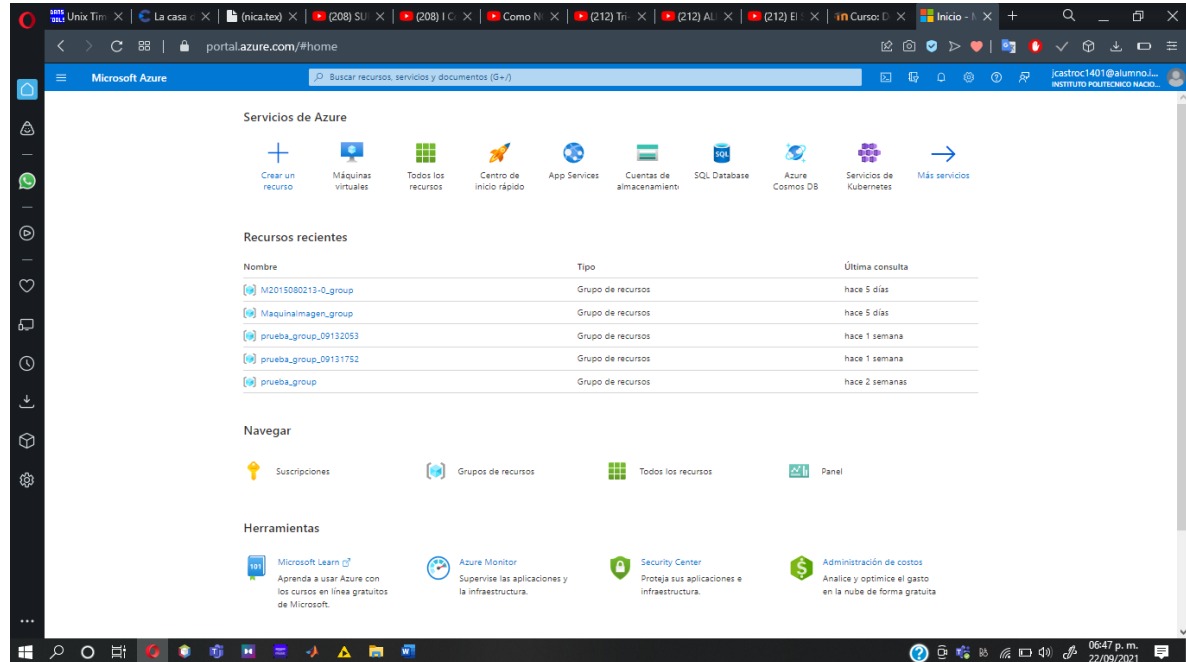
## 2. Pruebas de escritorio

Primer paso, vamos a crear la maquina virtual en la plataforma de Microsoft Azure:

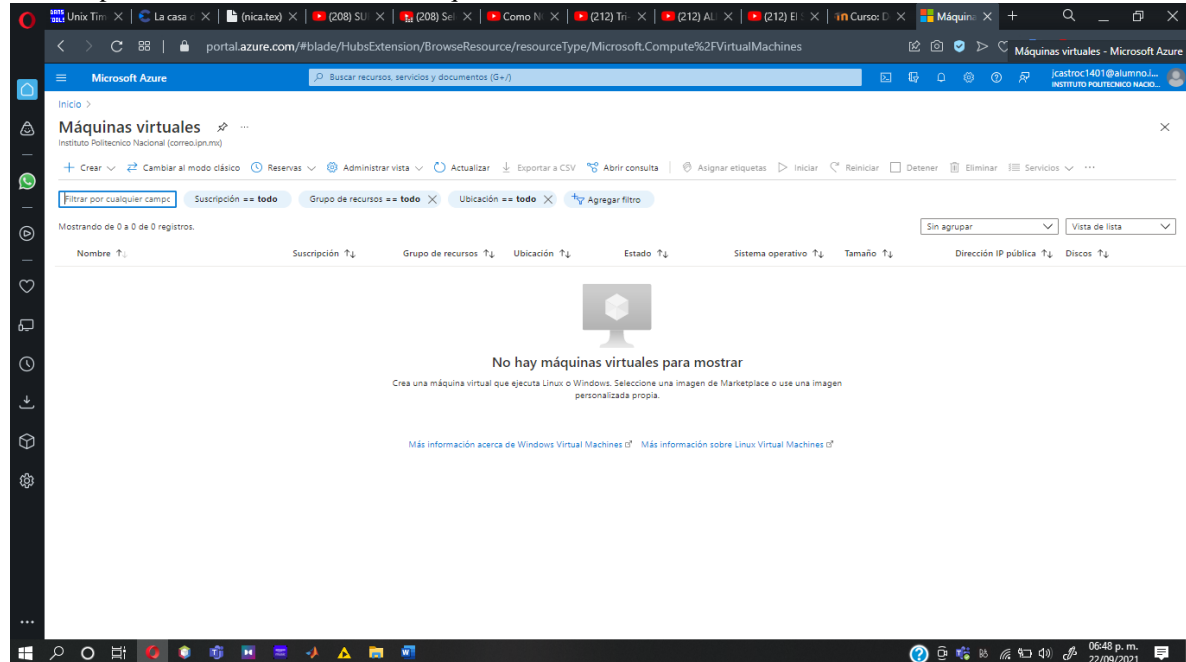
Ingresar al portal de Azure en la siguiente URL:

<https://azure.microsoft.com/es-mx/features/azure-portal/>

1. Dar click al botón "Iniciar sesión".

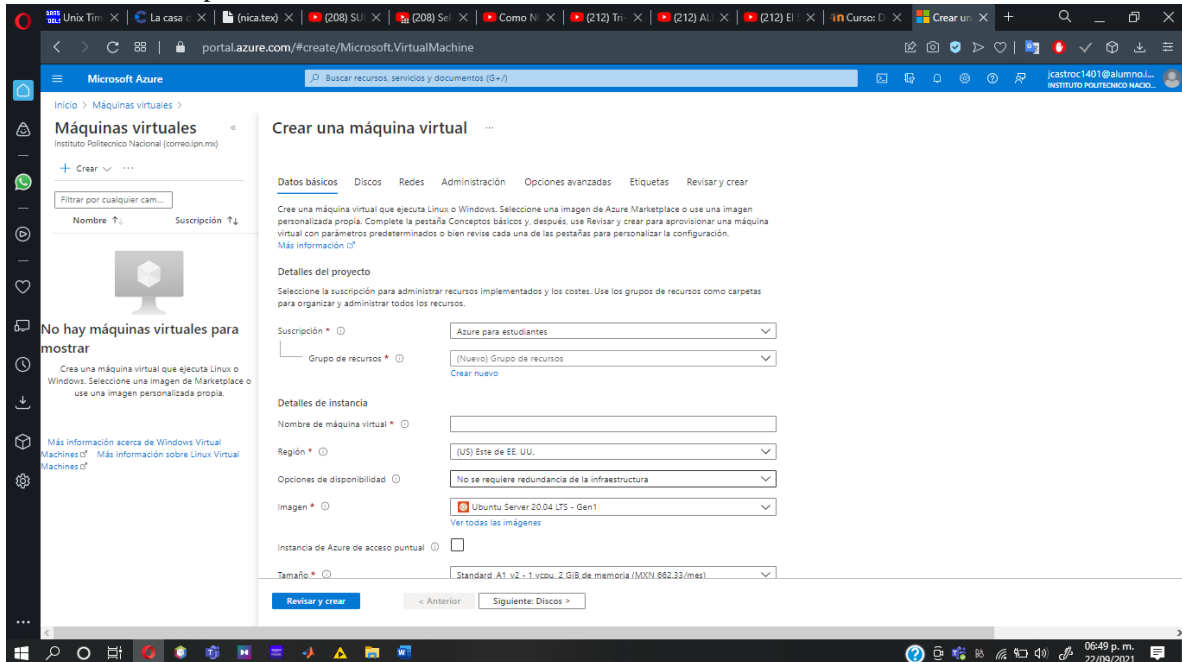


2. En el portal de Azure seleccionar "Máquinas virtuales".



3. Seleccionar la opción "+Crear".

4. Seleccionar la opción "+Virtual machine"

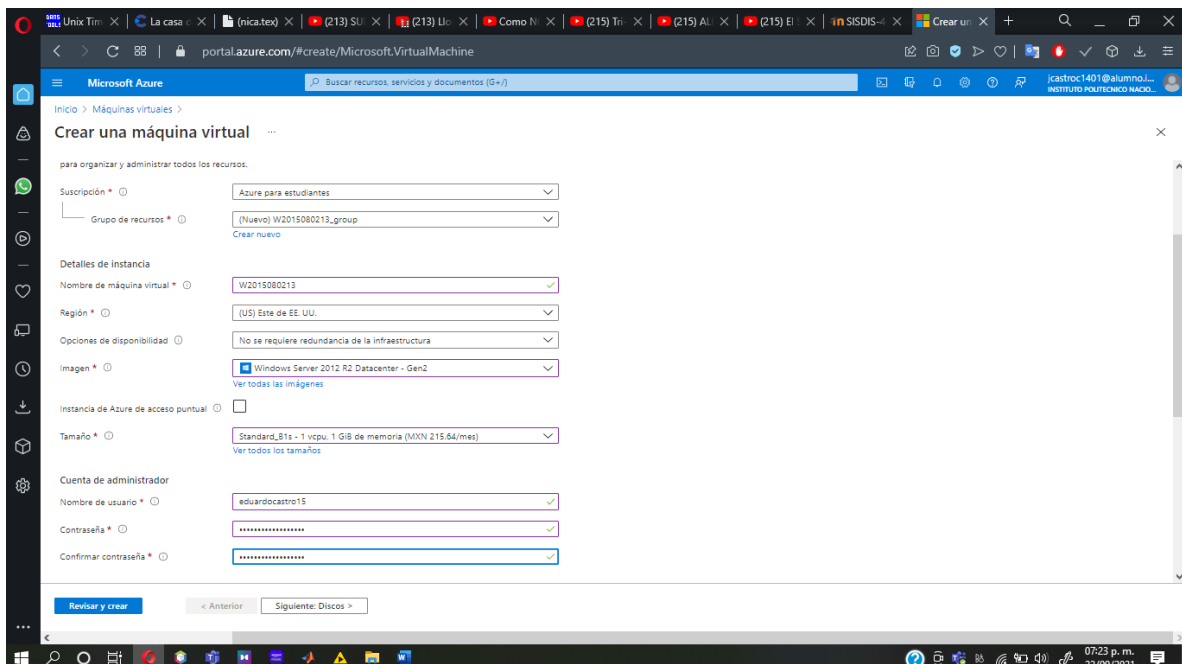


5. Seleccionar la región dónde se creará la máquina virtual. Notar que el costo de la máquina virtual depende de la región.

6. Seleccionar la imagen, en este caso vamos a seleccionar Windows Server 2012.

7. Seleccionar el tamaño de la máquina virtual, en este caso vamos a seleccionar una máquina virtual con al menos 2 GB de memoria.

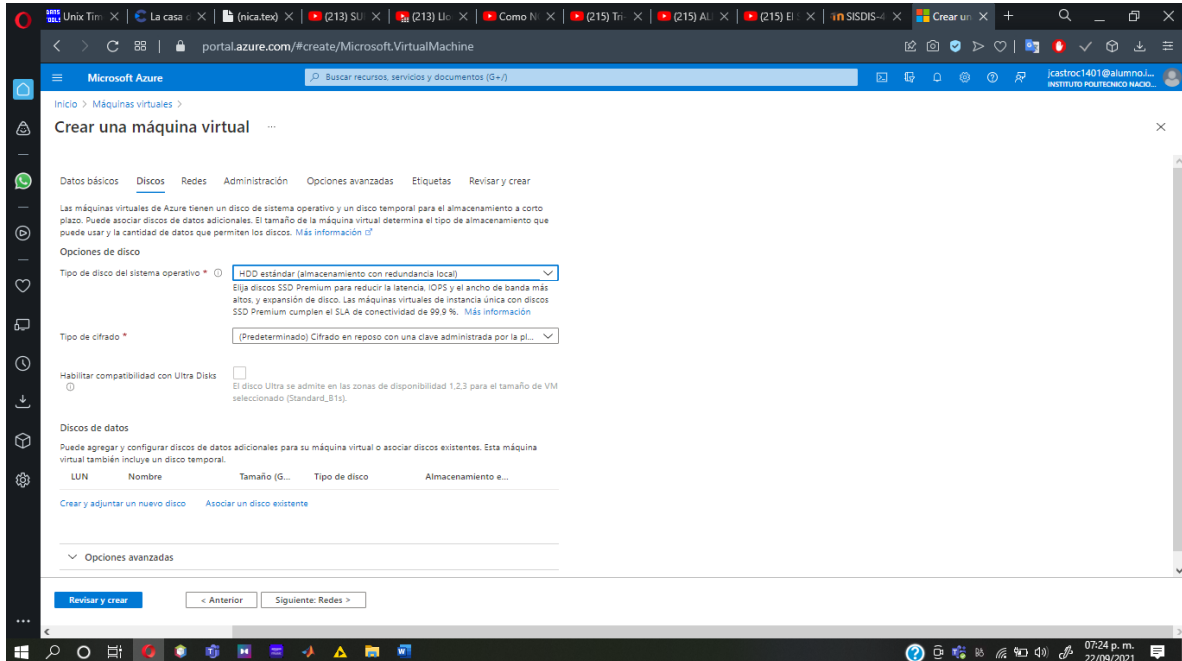
8. Ingresar el nombre del usuario administrador y la contraseña.



9. En las "Reglas de puerto de entrada" se deberá dejar abierto el puerto 3389 para utilizar Remote Desktop Protocol (RDP).

10. Dar click en el botón "Siguiente: Discos>"

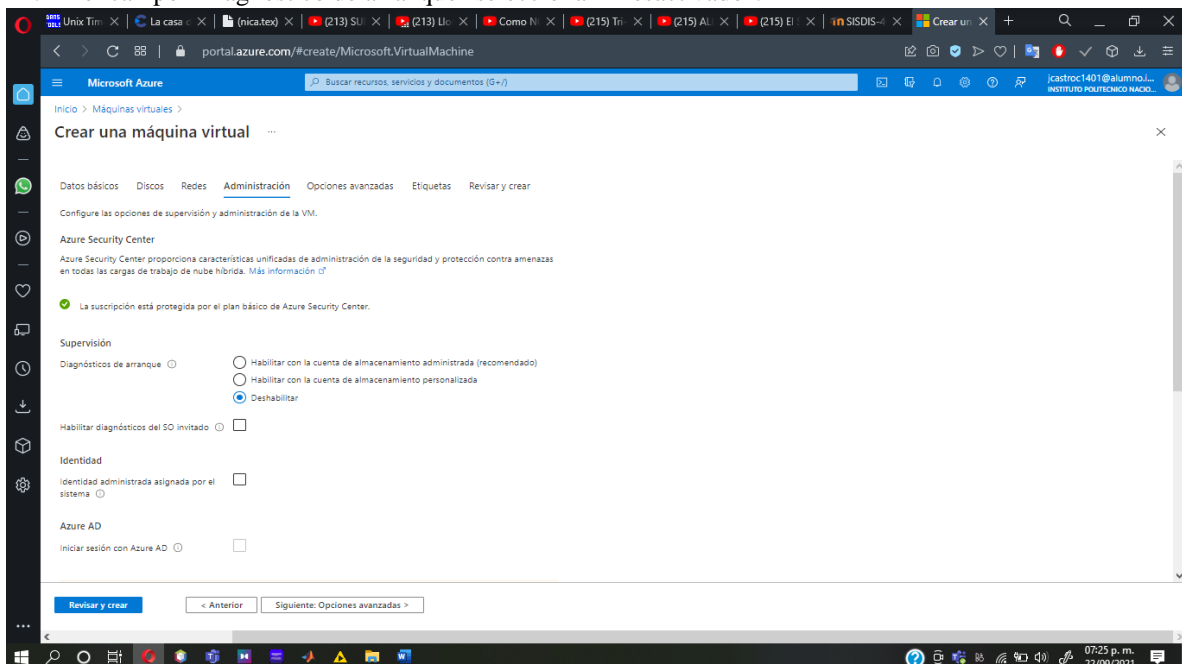
11. Seleccionar el tipo de disco de sistema operativo, en este caso vamos a seleccionar HDD estándar.



12. Dar click en el botón "Siguiente: Redes>"

13. Dar click en el botón "Siguiente: Administración>"

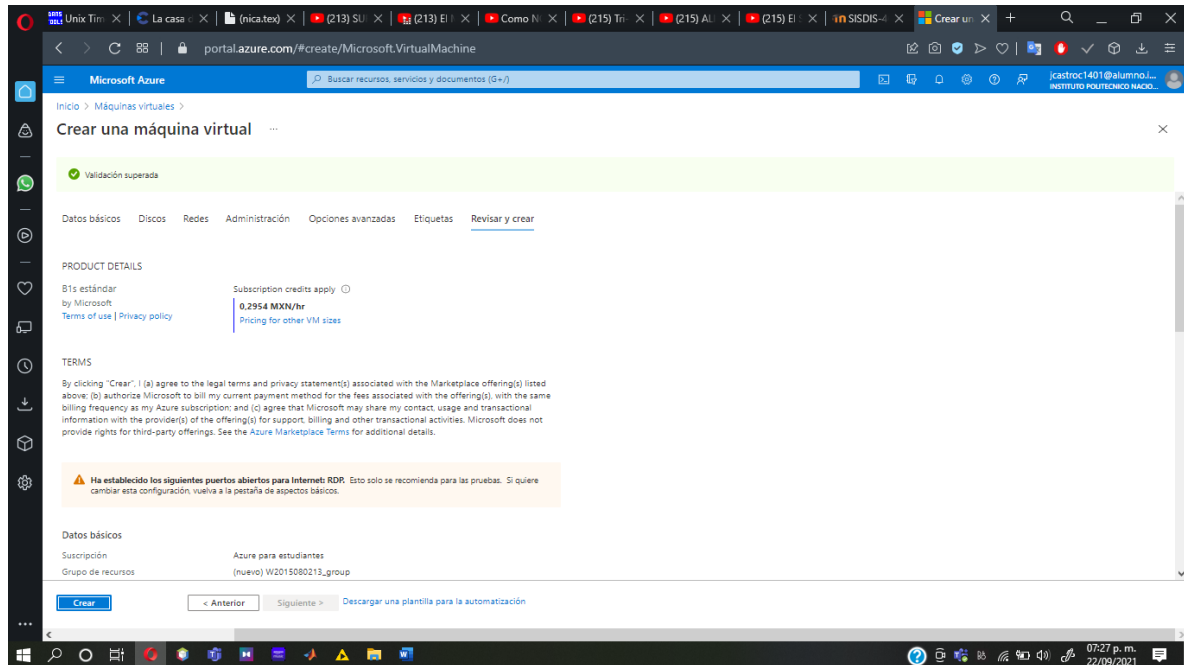
14. En el campo "Diagnóstico de arranque" seleccionar "Desactivado".



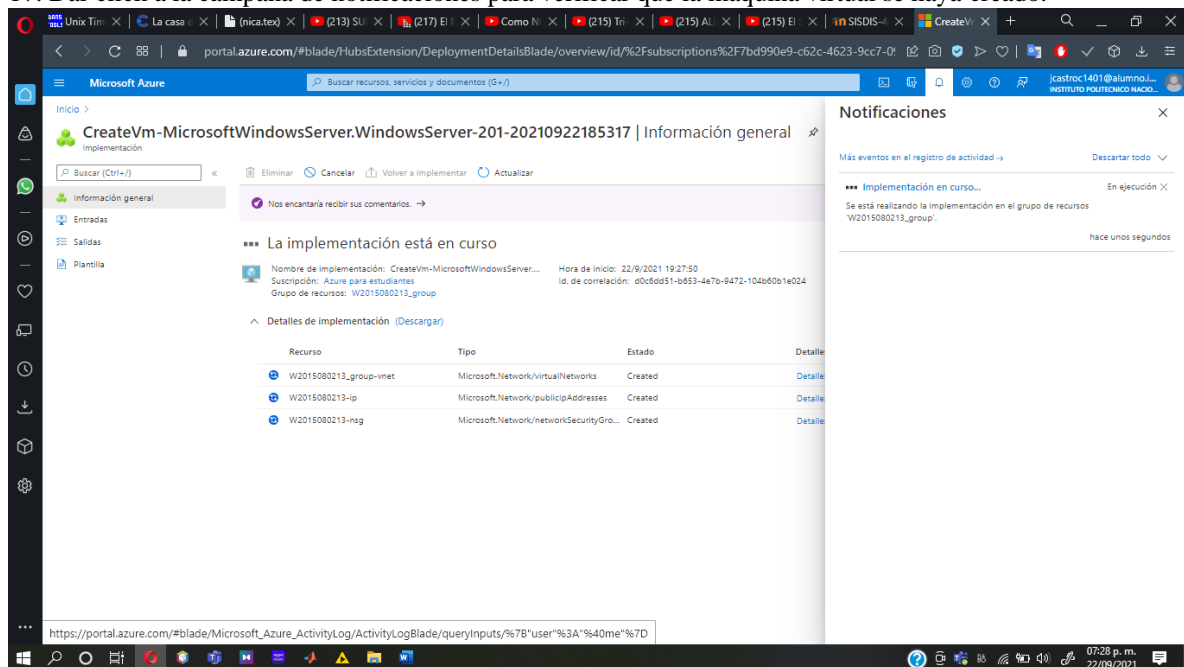


15. Dar click en el botón "Revisar y crear".

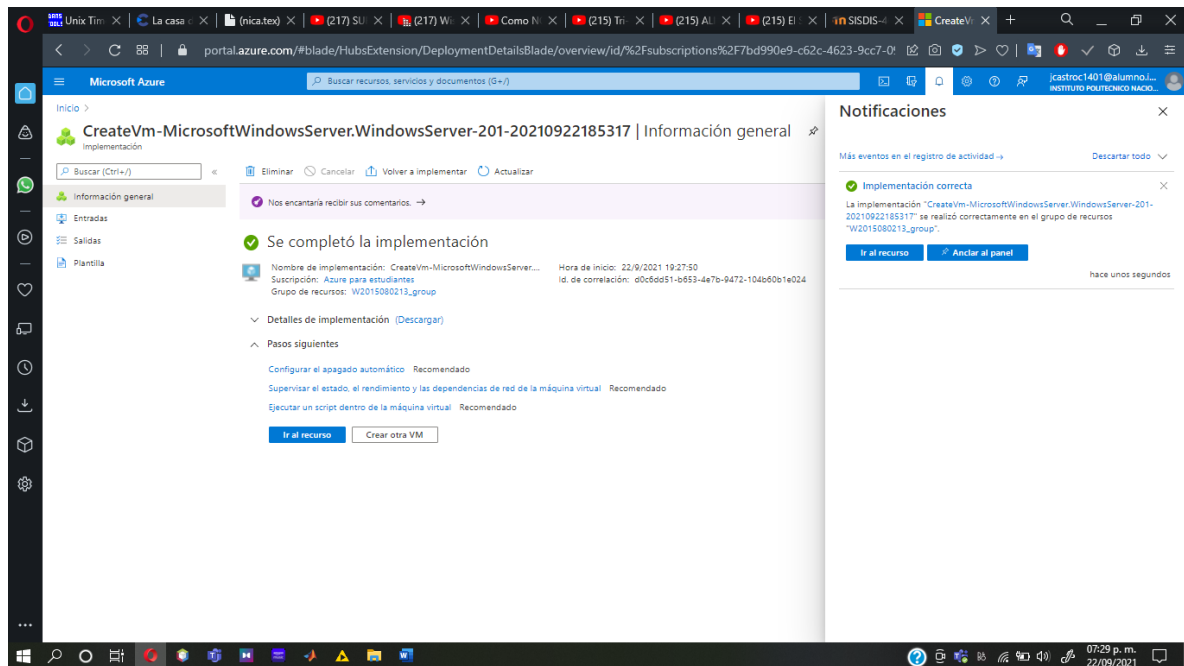
16. Dar click en el botón "Crear".



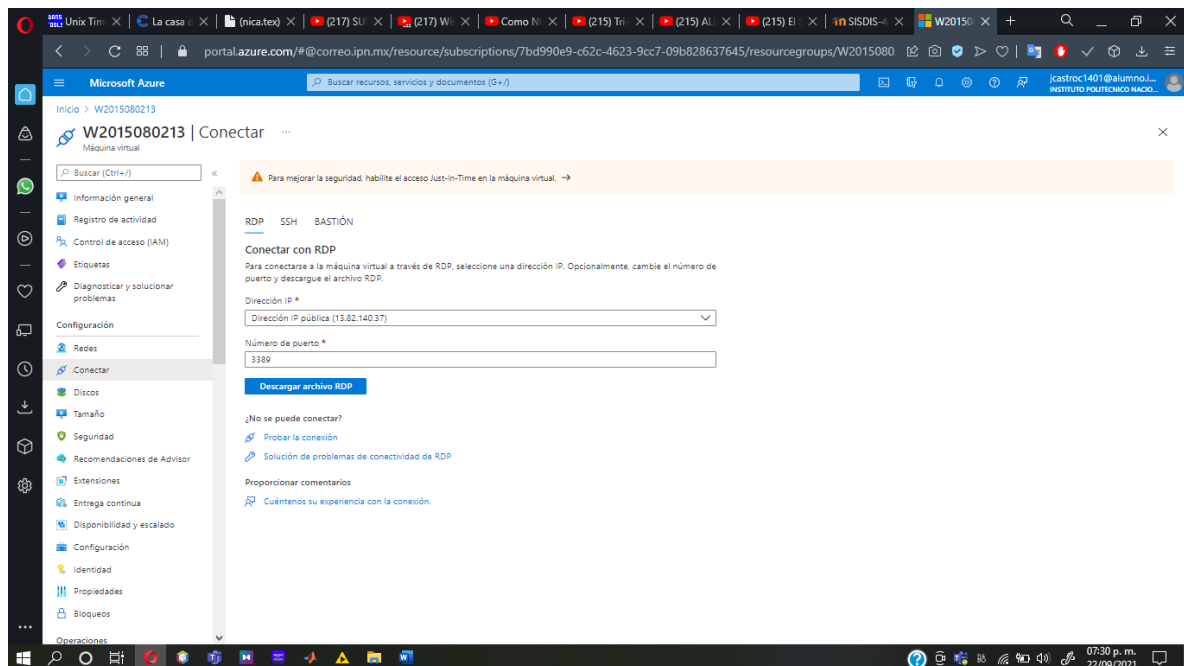
17. Dar click a la campana de notificaciones para verificar que la maquina virtual se haya creado.



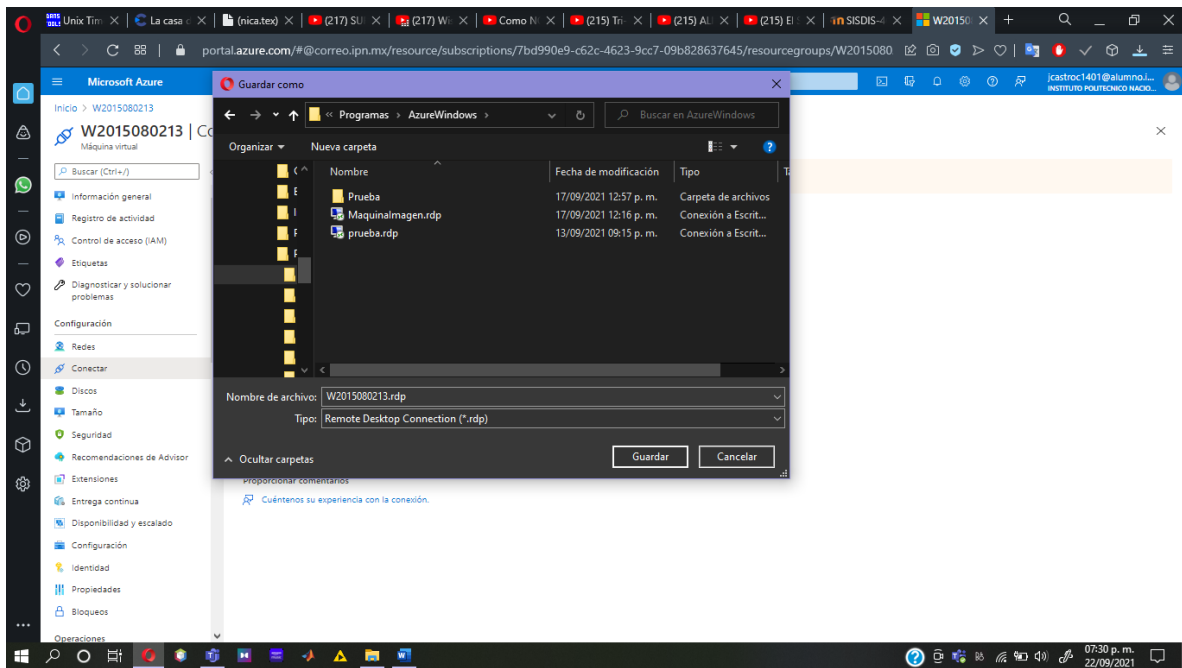
18. Dar click en el botón "Ir al recurso".



19. Seleccionar la opción "Conectar". Seleccionar "RDP".



20. Dar click en el botón "Descargar archivo RDP".



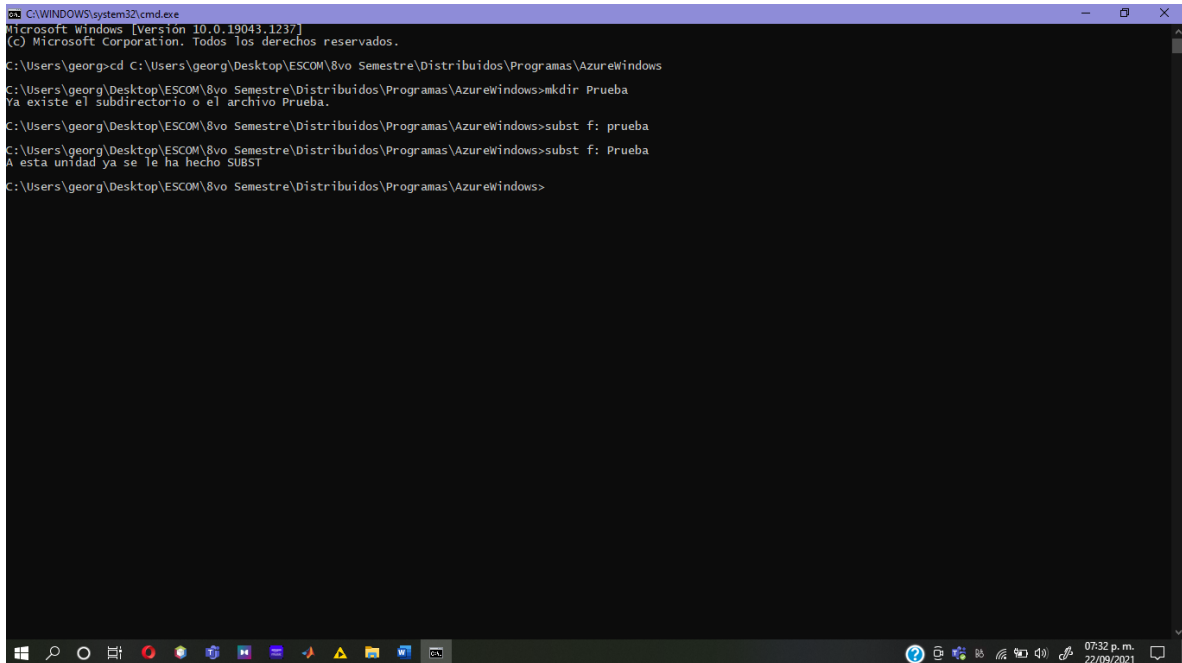
21. Ejecutar "cmd" en la computadora local.

22. Vamos a crear un directorio en la computadora local. La máquina virtual recién creada va a ver este directorio como un disco lógico. Por ejemplo, el directorio se llamará "prueba". Ejecutar el siguiente comando en la ventana de Símbolo del sistema:

*mkdir prueba*

23. Ahora vamos a crear un disco lógico como alias del directorio creado. Ejecutar el siguiente comando:

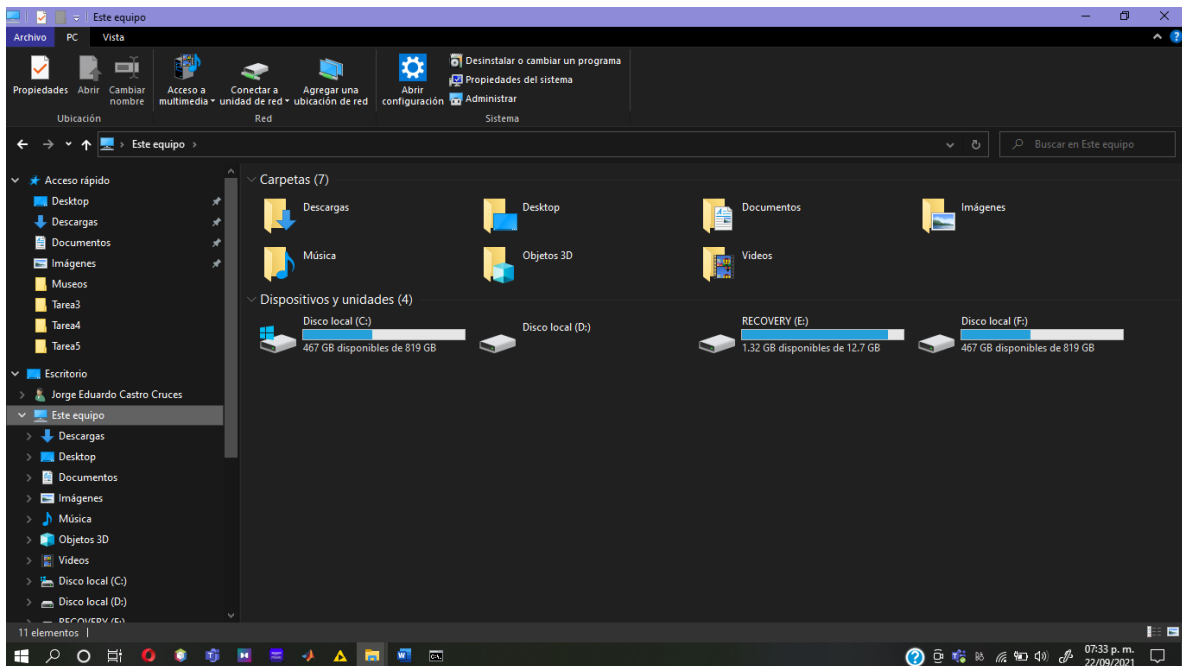
*subst f: prueba*



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.19043.1237]
(c) Microsoft Corporation. Todos los derechos reservados.

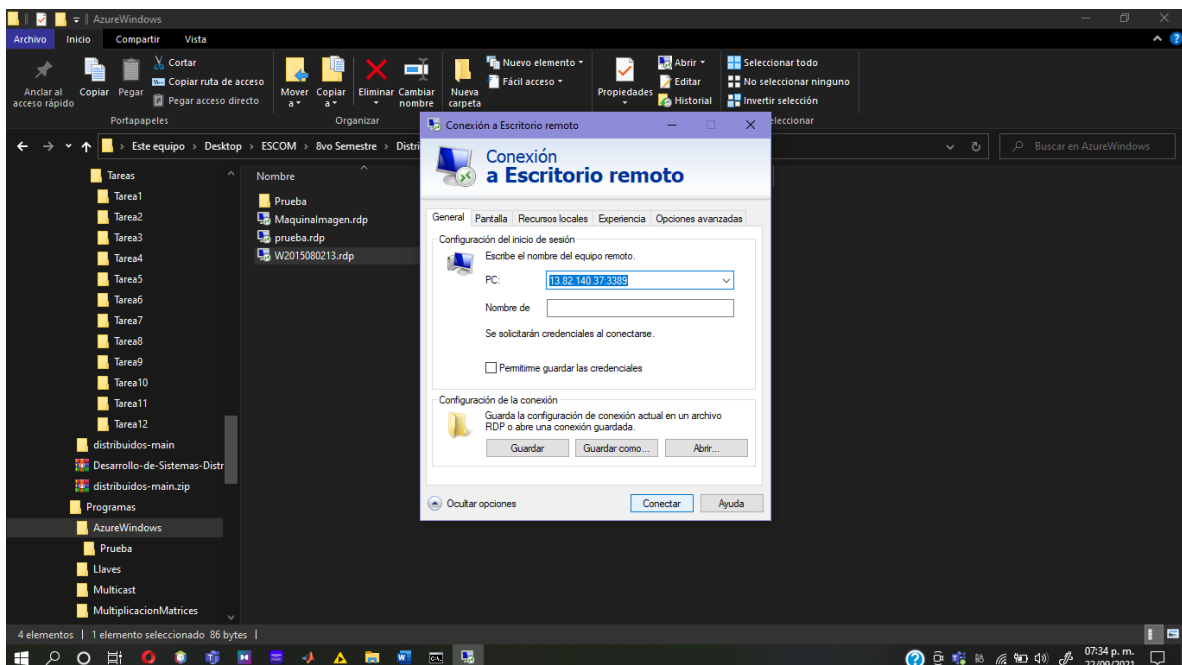
C:\Users\georg>cd C:\Users\georg\Desktop\ESCOM\8vo Semestre\Distribuidos\Programas\AzureWindows
C:\Users\georg\Desktop\ESCOM\8vo Semestre\Distribuidos\Programas\AzureWindows>mkdir Prueba
Ya existe el subdirectorio o el archivo Prueba.
C:\Users\georg\Desktop\ESCOM\8vo Semestre\Distribuidos\Programas\AzureWindows>subst f: prueba
C:\Users\georg\Desktop\ESCOM\8vo Semestre\Distribuidos\Programas\AzureWindows>subst f: Prueba
A esta unidad ya se le ha hecho SUBST
C:\Users\georg\Desktop\ESCOM\8vo Semestre\Distribuidos\Programas\AzureWindows>
```

Podemos ver que el disco lógico aparece en el explorador de archivos de Windows.

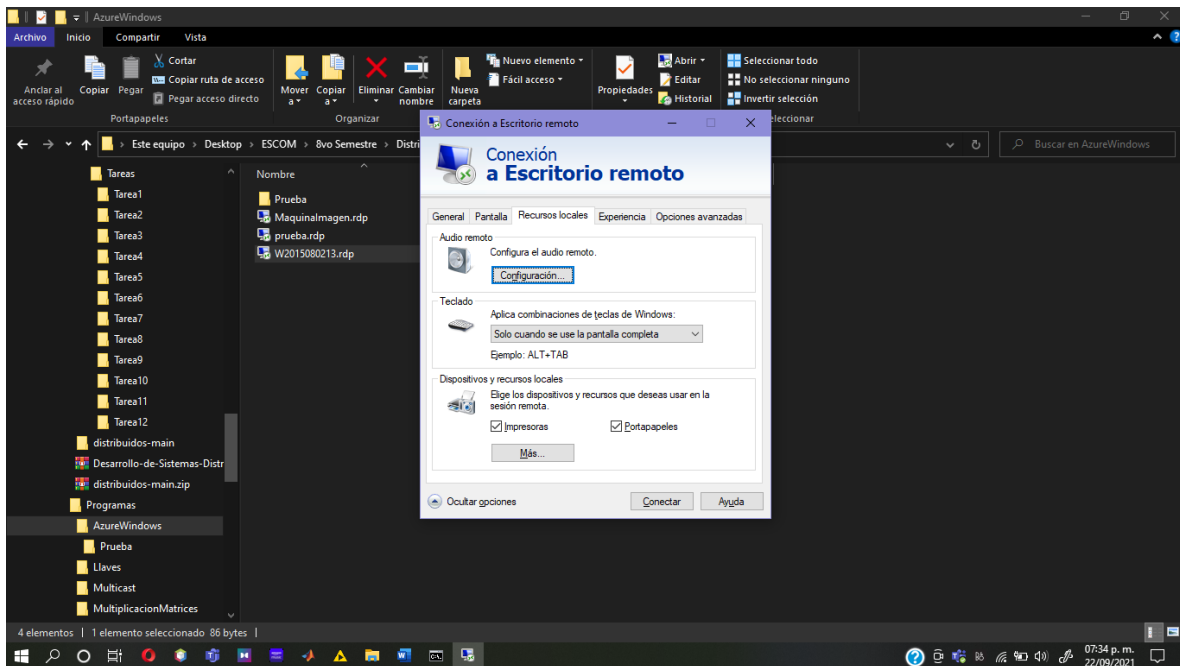


24. Buscar el archivo de conexión en la carpeta de descargas (un archivo con el nombre de la máquina virtual y la extensión ".rdp").

25. Dar click derecho al archivo de conexión y seleccionar "Modificar".



26. Seleccionar la pestaña "Recursos locales".

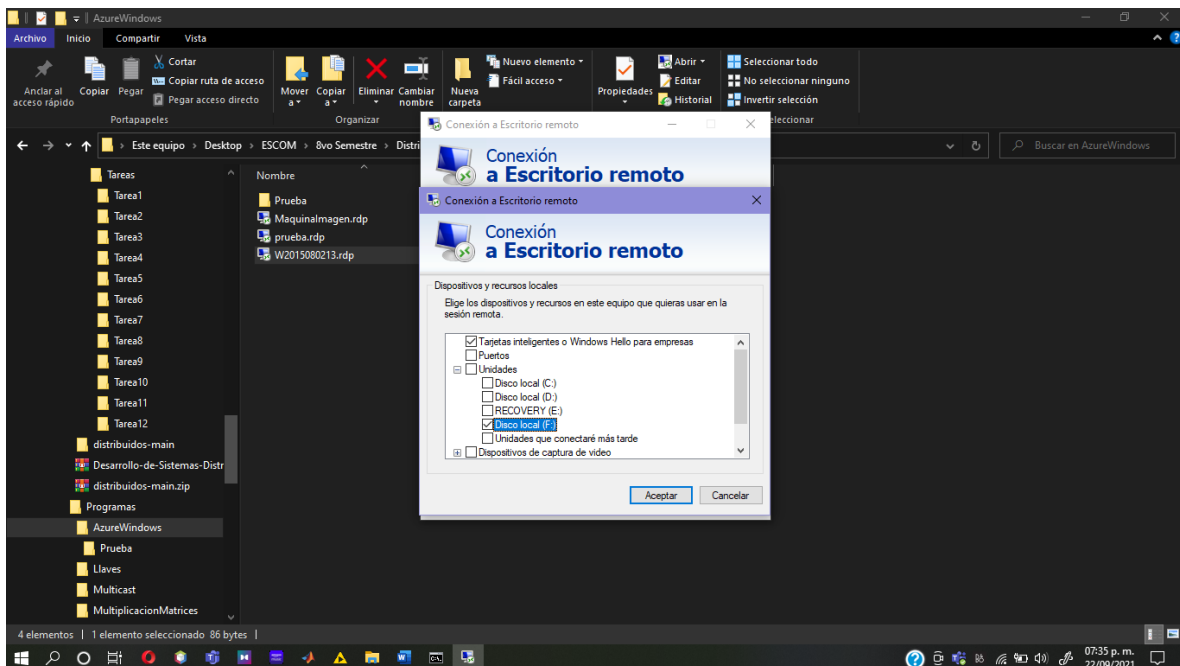


27. Dar click en el botón "Mas..."

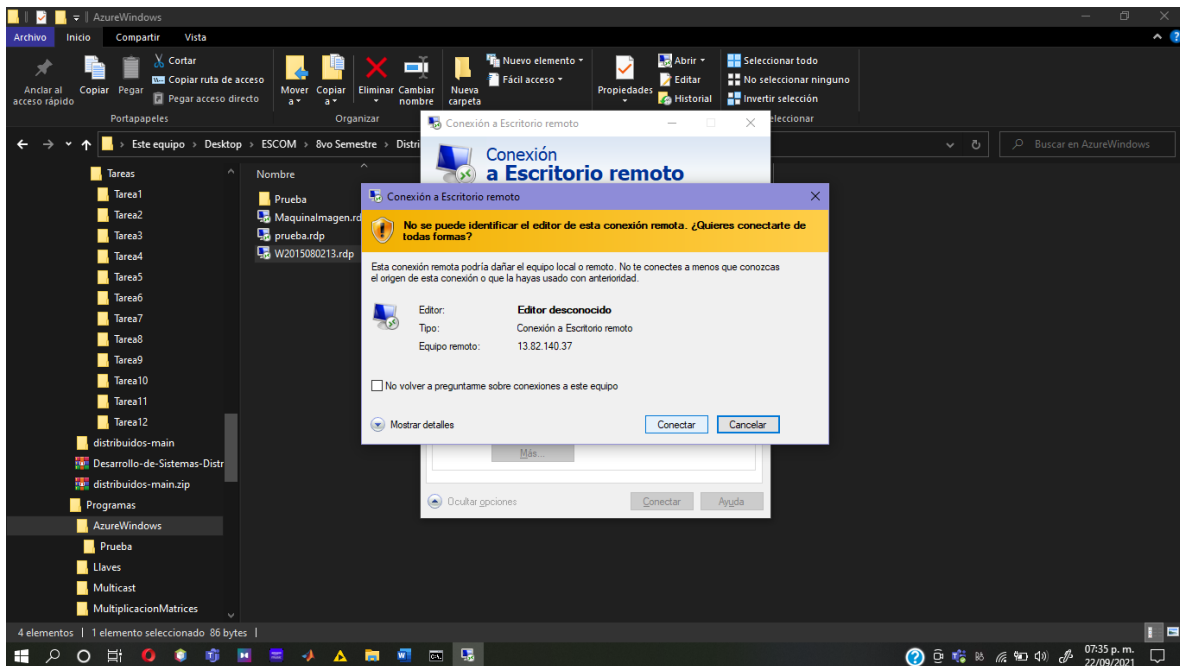
28. Abrir la sección "Unidades".

29. Marcar la casilla "Windows (F:)"

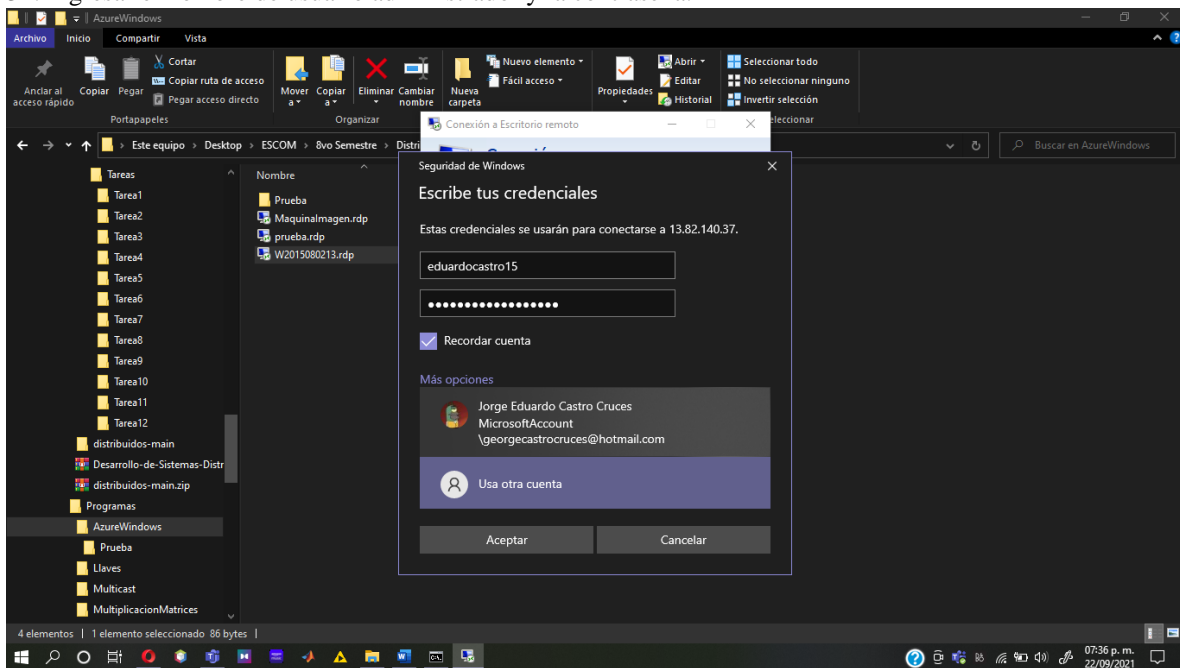
30. Dar click en el botón "Aceptar".



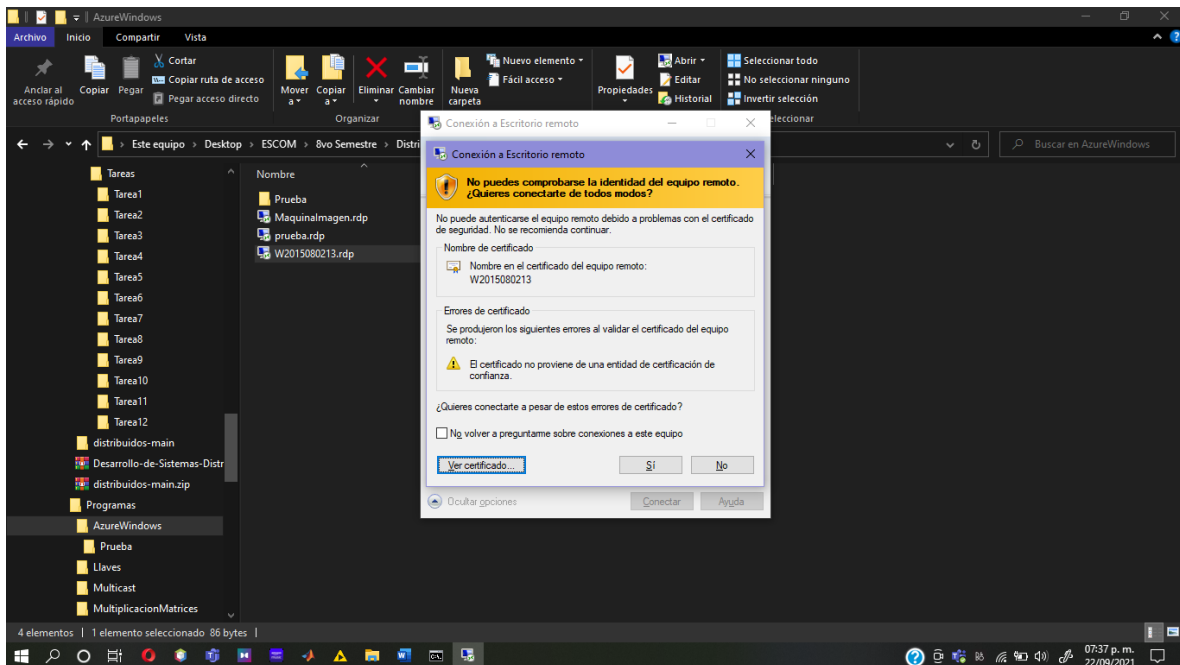
31. Dar click en el botón "Conectar" en la pantalla de advertencia.



32. Ingresar el nombre de usuario administrador y la contraseña.

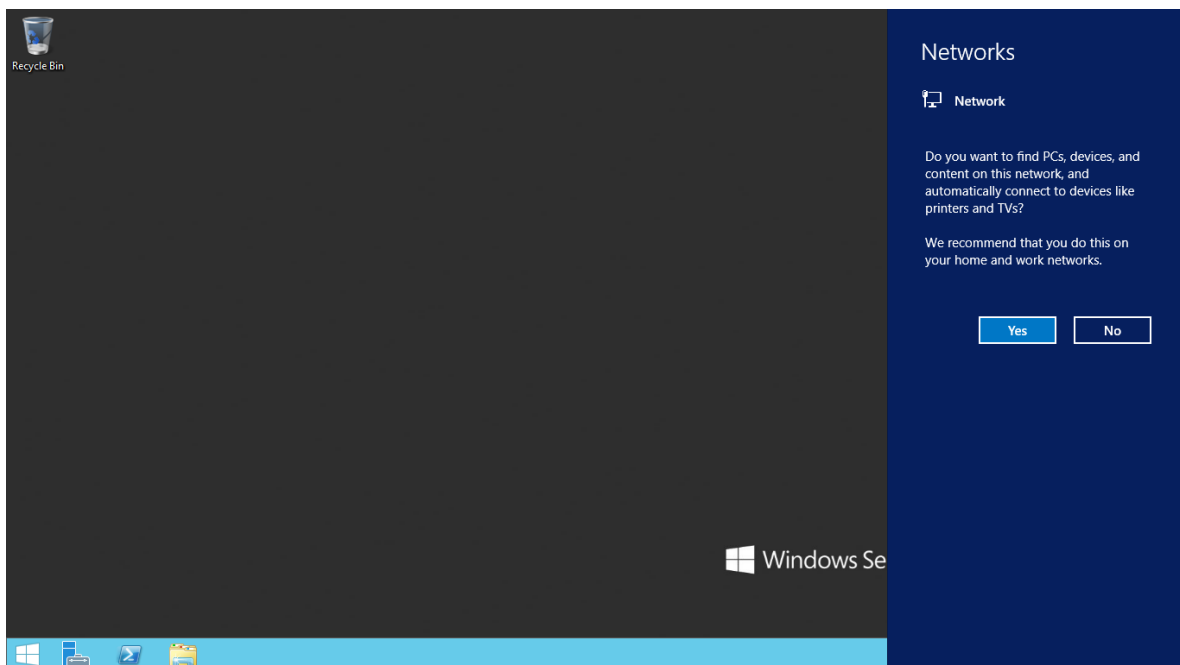


33. Dar click en el botón "Sí" en la ventana de advertencia. Entonces se abrirá una ventana de escritorio remoto, la cual nos dará acceso al escritorio de la máquina virtual.



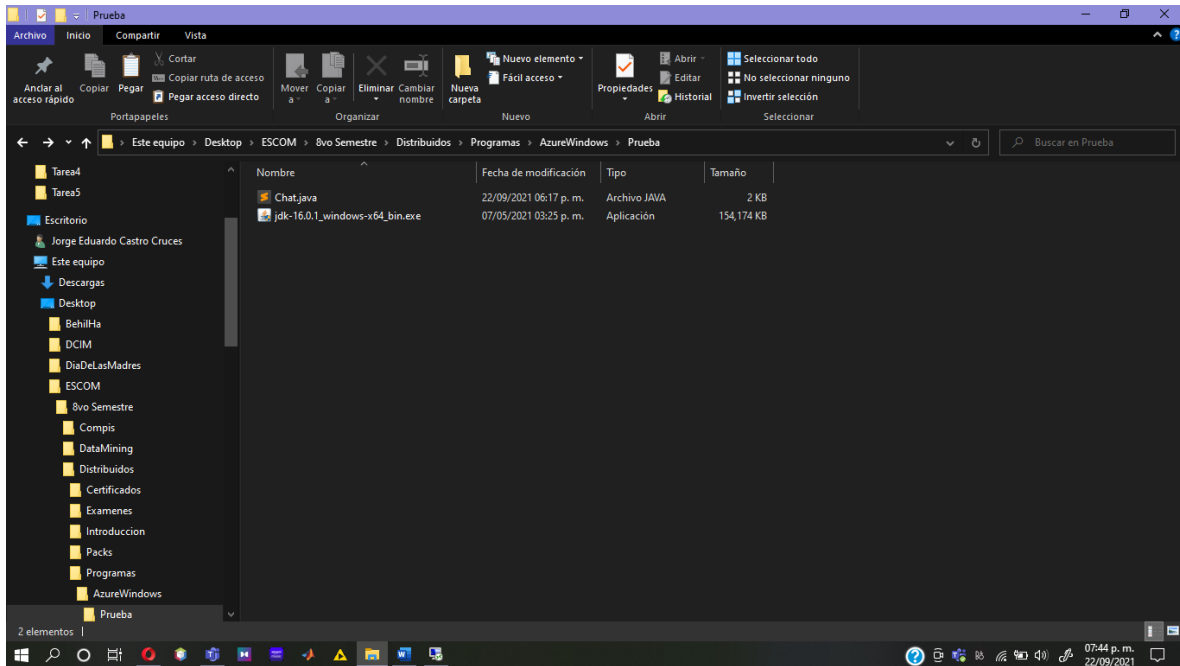
34. Configurar los parámetros de privacidad y dar click en el botón "Accept".

35. En la ventana "Networks" dar click en el botón "No".



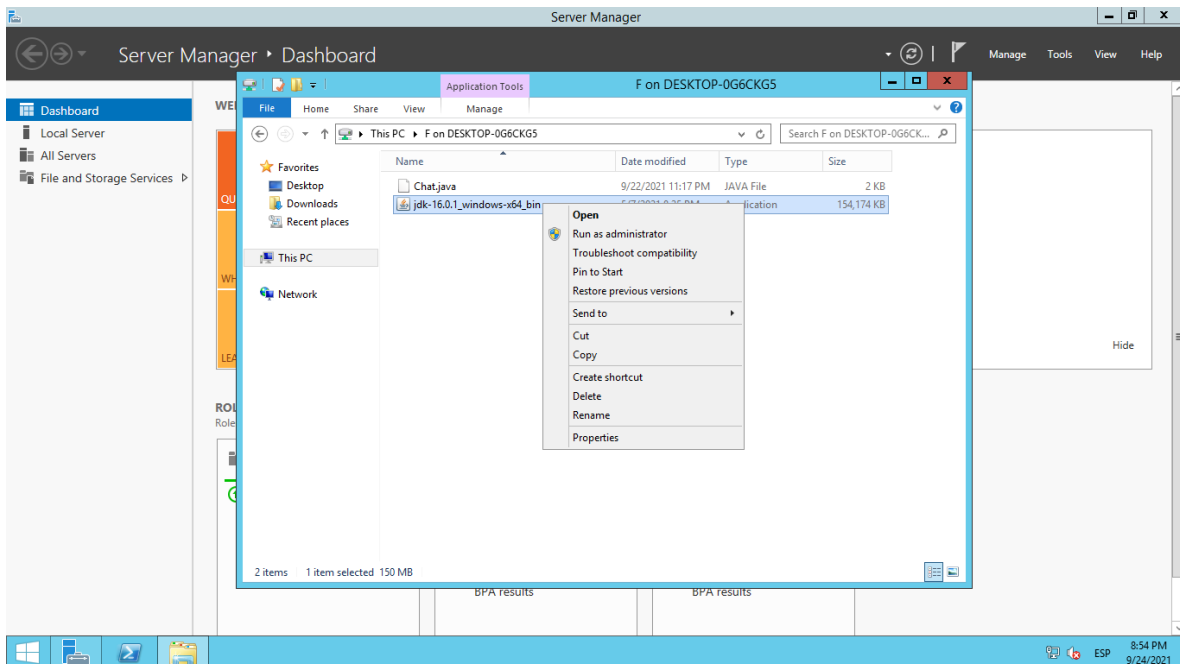


36. Para ver el disco lógico creado en el paso 23, abrir el explorador de Windows de la máquina virtual. Entonces para enviar archivos desde la computadora local a la máquina virtual se deberá colocar los archivos en el directorio creado en el paso 22, y para enviar archivos desde la máquina virtual a la computadora local se deberá colocar los archivos en el disco F de la máquina virtual.

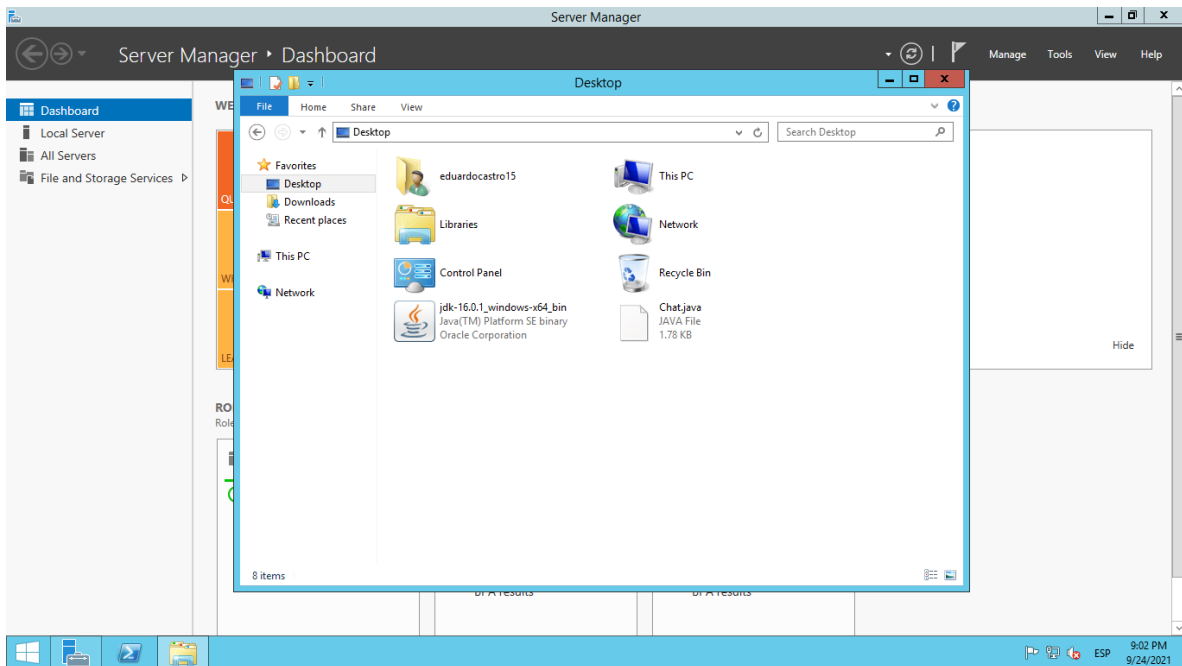


**Nota.** El teclado local podría no coincidir con la configuración del teclado de la maquina remota.

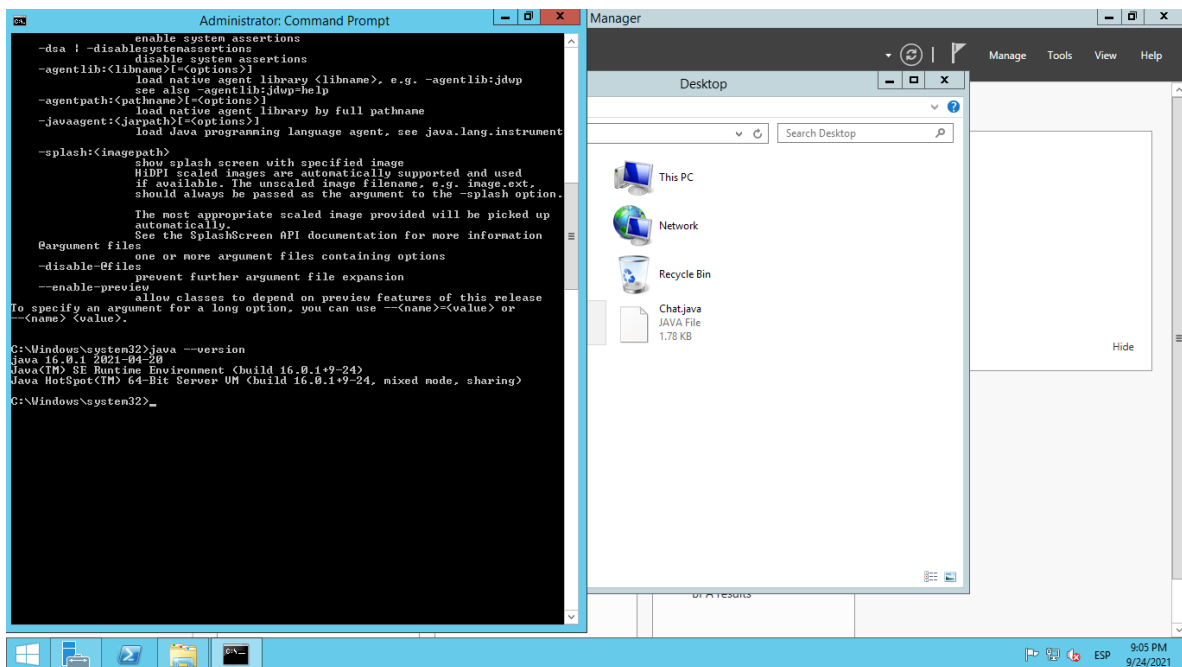
37. Una vez establecida la conexión, debemos dirigirnos al disco compartido para poder copiar el instalador de JDK 16 y el código del Chat.java al escritorio de la maquina virtual.



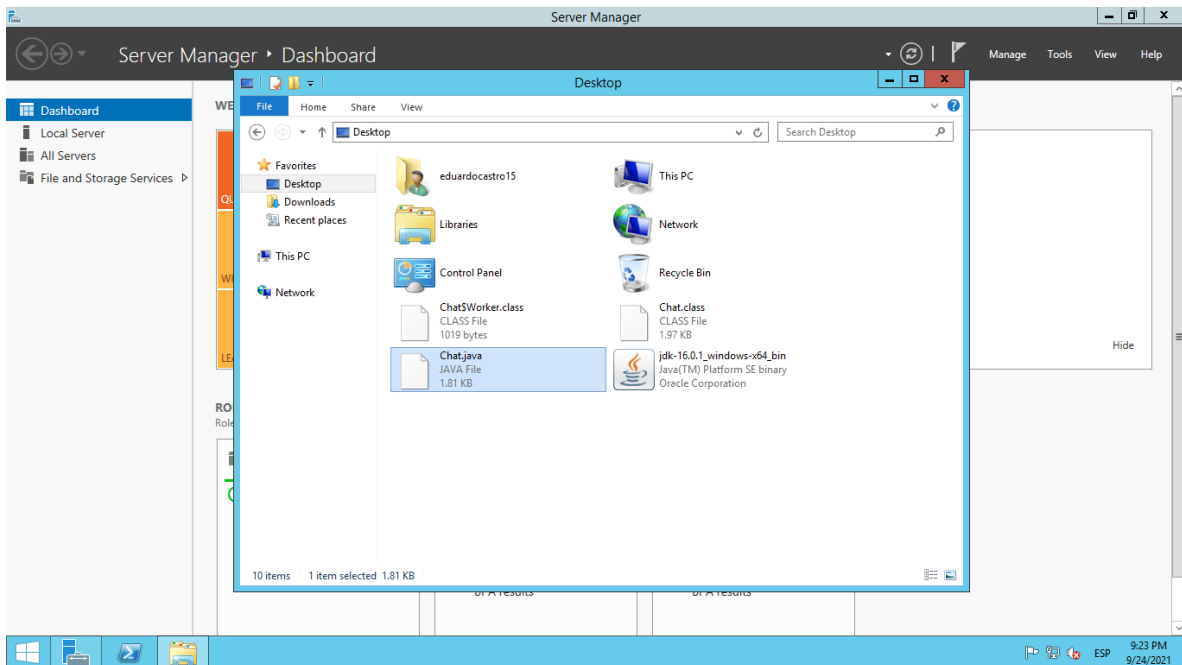
38. Ya copiados ambos archivos, procedemos a la instalación del programa.



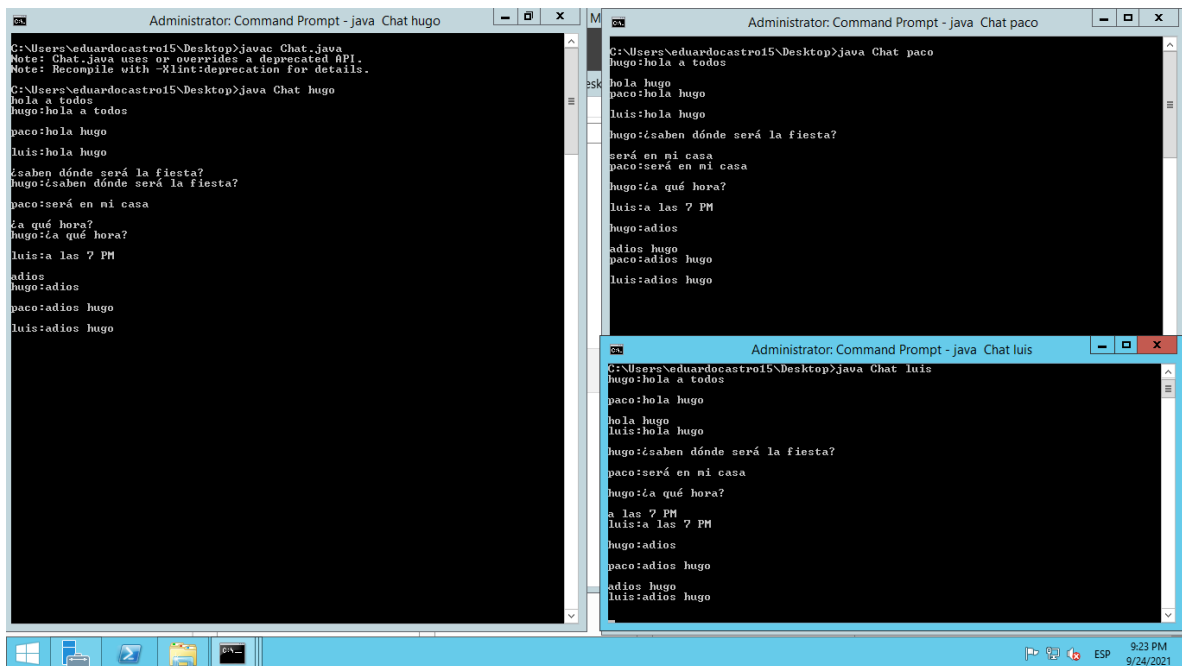
39. Pasamos a verificar la versión de java instalada mediante consola.



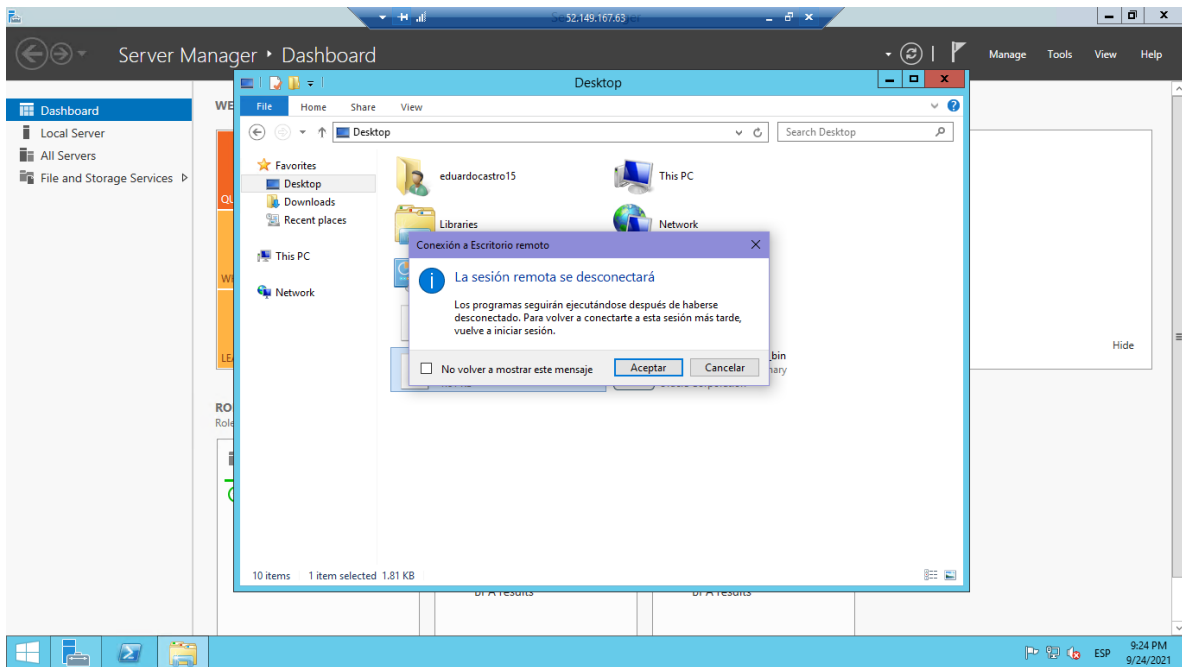
40. Procedemos a la compilación del archivo Chat.java, el cual genera dos archivos: Chat\$Worker.class y Chat.class.



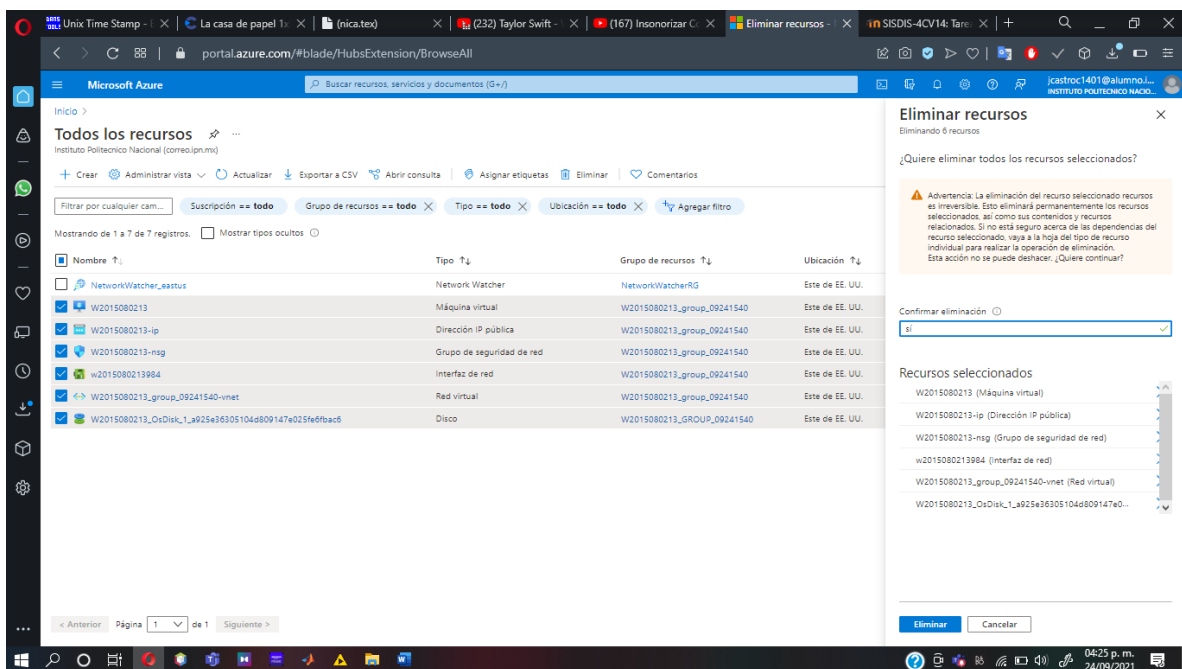
41. Hecho lo anterior, podemos ejecutar en 3 consolas diferentes el chat multicast, con la conversión y el formato propuesto por el profesor.



42. Para desconectarse de la máquina virtual, dar click en el botón "X" del escritorio remoto. Notar que al cerrar el escritorio remoto la máquina virtual sigue ejecutando.



43. Por último, vamos a borrar la maquina virtual, para evitar costos extra.



### 3. Conclusiones

En esta cuarta práctica del curso de sistemas distribuidos sí comprendió de mejor forma el tema visto en clase de comunicaciones multicast, las cuales nos permiten enviar transmitir mensajes a más de un usuario o cliente.

En esta práctica utilizamos el puerto 40,000 y la IP 230.0.0.0 la cual como sabemos es una IP reservada para comunicaciones multi cast.

finalmente se lograron los objetivos de la práctica los cuales eran crear un chat multicast para 3 usuarios Hugo, Paco, Luís y éstos pudieran tener una conversación mediante cadenas de caracteres utilizando como codificación la IBM850, la cual nos permite enviar caracteres especiales en este caso las vocales acentuadas sin que sufrirá modificación alguna ya que estamos ejecutando el programa mediante consola y eso nos restringía en cuestión de codificación.

Por último reforzamos el conocimiento en la implementación de máquinas virtuales utilizando la plataforma de Azure De Microsoft, con un sistema operativo de Windows 2012 Server y con las siguientes características: 8 Gigas de RAM, 2 CPU's virtuales, y un disco SSD estándar, todo esto con el fin de que nuestro programa instalación del JDK 16 fuese lo más rápido posible ya que sabemos Que Windows consume muchos recursos gráficamente hablando y más siendo una máquina virtual nos íbamos a llevar mucho tiempo de procesamiento, por eso preferí utilizar estas características en la máquina virtual la cual me permitió hacer la instalación del programa un minuto, y al no tardarme más de una hora utilizando la máquina virtual no hubo costo alguno por parte de la plataforma.