



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Desarrollo de Sistemas Distribuidos



Tarea 4

Chat Multicast

PROFESOR: Pineda Guerrero Carlos

EQUIPO 12

- Ramírez Galindo Karina
- Toledo Espinosa Cristina Aline
- Vázquez Hernández Alan Mauricio

GRUPO: 4CV11

Fecha de realización: 17 - marzo - 2022

Fecha de entrega: 18 – marzo - 2022

Contenido

Introducción.....	1
Transmisión de multicast.....	1
Direcciones Multicast.....	1
Socket Datagrama.....	2
Cliente – Servidor Multicast	2
Desarrollo	3
Funcionamiento.....	5
Creación de la máquina virtual.....	5
Configuración de la máquina virtual.....	15
Compilación y ejecución del programa.....	27
Concusiones.....	35
Ramírez Galindo Karina.....	35
Toledo Espinosa Cristina Aline.....	35
Vázquez Hernández Alan Mauricio	35
Referencias	36

Introducción

Transmisión de multicast

La transmisión de multicast está diseñada para conservar el ancho de banda de las redes IPv4. Reduce el tráfico al permitir que un host envíe un único paquete a un conjunto seleccionado de hosts que forman parte de un grupo multicast suscrito. En la figura 1 se muestran los hosts pertenecientes a dicho grupo. [1]

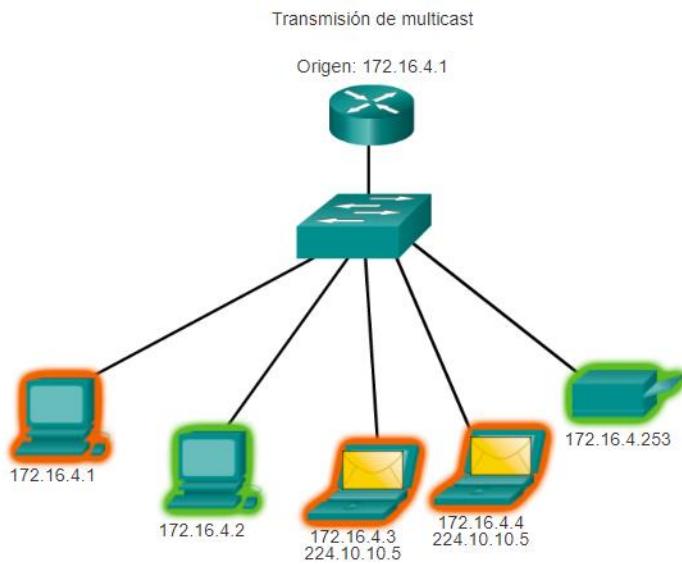


Figura 1 Transmisión Multicast

Direcciones Multicast

El direccionamiento grupal es una de las propiedades clave de la multidifusión IP, ya que es fundamental para el funcionamiento de este concepto de comunicación.

El rango de direcciones de las redes IP es de 224.0.0.0 a 239.255.255.255 (o bien FF00::/8), también conocido como espacio de direcciones clase D, que está reservado a tal fin.

De forma general, el acceso a los grupos IP multicast está regido por los routers de red involucrados y el protocolo de administración de grupos de Internet (IGMP). Para ello, el protocolo, que pertenece a la familia de protocolos de Internet, proporciona varios tipos de mensajes. Los servidores pueden utilizarlos para informar al router local sobre una solicitud de pertenencia al grupo y los routers, para recibir y reenviar las secuencias IP multicast que correspondan. El punto de inicio para la comunicación IGMP es siempre el router desde el que se conecta a la red el emisor

del multicast. El emisor reenvía los paquetes de un flujo de datos una sola vez, indicando la dirección del grupo de multidifusión como dirección de destino y sin saber realmente a cuántas terminales ha llegado. [2]

Socket Datagrama

- Los datos son enviados en un paquete a la vez.
- No se requiere establecer una conexión.
- El protocolo UDP (User Datagram Protocol) es el más utilizado para sockets sin conexión.

Las características de los sockets sin conexión son las siguientes:

- No requieren un canal dedicado de comunicación.
- No se garantiza la integridad de los datos enviados.
- Los paquetes son enviados y recibidos en diferente orden.
- Los paquetes pueden recibirse duplicados.
- Rápida recuperación de datos perdidos en el camino.
- No hay acknowledgement ni re-envío.
- Utilizados para broadcast y multicast.
- Utilizados para la transmisión de audio y video en tiempo real.
- Se implementan mayormente usando el protocolo UDP.

Cliente – Servidor Multicast

En el caso de la comunicación multicast, el servidor es el programa que envía mensajes a los clientes, por esta razón es necesario que los clientes invoquen la función *receive* antes que el servidor ejecute la función *send*. La comunicación multicast mostrada en la figura 2 se implementa mediante sockets sin conexión, por tanto, no se requiere que se establezca una conexión dedicada entre el servidor y el cliente. [3]



Figura 2 Cliente – Servidor Multicast

Desarrollo

A lo largo de esta práctica se desarrolla un programa en Java que implementa un chat utilizando comunicación multicast mediante datagramas. Dicho chat cumple con las siguientes características:

- Se ejecuta el programa en una máquina virtual con Windows Server 2012 en Azure.
- Se pasa como parámetro al programa el nombre del usuario que escribe en el chat.
- Para demostrar el programa se utilizan los siguientes usuarios: Juan, Pedro y Ana.

NOTA: Se solicita que los signos de interrogación y las letras acentuadas **deben** desplegarse correctamente en la ventana de comandos de Windows (se sugiere, utilizar la codificación de texto que corresponda).

Para el intercambio de mensajes se crean los métodos a continuación (figura 3 envío, figura 4 recepción). Para el caso de envío se le deben pasar como parámetros: el mensaje a enviar, la IP del grupo multicast y el puerto por el que se transmitirá.

```
public static byte[] recieveMessage(MulticastSocket socket, int messageLength) throws IOException {
    byte[] buffer = new byte[messageLength];
    DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
    socket.receive(packet);
    return packet.getData();
}
```

Figura 3 Método para enviar mensajes Multicast

Para recibir los mensajes se deben pasar como parámetros: el nombre del socket y la longitud del mensaje.

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

Figura 4 Método para recibir mensajes Multicast

Se crea un thread que actúa como cliente multicast (figura 5), el cual recibe los mensajes del resto de los nodos y estos a su vez serán retransmitidos mediante la dirección de grupo “230.0.0.0” por el puerto 50000. Adicionalmente para que al momento de ejecutar en consola aparezcan los caracteres especiales (comas, acentos, signos, etc) se utiliza el juego de caracteres CP850 de 8 bits, el cual contiene 256 posibles caracteres.

```
private static String ip = "230.0.0.0";
private static final int MESSAGE_LENGTH = 1024;

static class Worker extends Thread {

    @Override
    public void run() {
        //String message;
        PrintStream message;
        MulticastSocket socket = null;
        NetworkInterface nic = null;
        while(socket == null) //Tries until a MulticastSocket is created
            try {
                nic = NetworkInterface.getByName("em1");//network interface name
                socket = new MulticastSocket(50000);
                socket.setReuseAddress(true);//address reuse
            } catch (IOException e1) {
                e1.printStackTrace();
            }
        InetAddress address = null;
        SocketAddress group = null;
        while(address == null) //Tries until a InetAddress is created
            try {
                address = InetAddress.getByName("230.0.0.0");
                group = new InetSocketAddress(address,50000);//group and port to transmit to (50000)
            } catch (UnknownHostException e1) {
                e1.printStackTrace();
            }
        try {
            //socket.joinGroup(group); //The socket joins the group. (This does not work on JDK 14 onwards)
            socket.joinGroup(group, nic);//IGMP membership report --> joining address 230.0.0.0 by selected NIC
        } catch (IOException e1) {
            e1.printStackTrace();
        }

        while(true) //Infinitely waits for messages and prints them
            try {
                byte[] buffer = receiveMessage(socket, MESSAGE_LENGTH);
                message = new PrintStream(System.out,true,"Cp850");
                message.print("\n"+new String(buffer).trim()+"\n");
            } catch (IOException e) {
                e.printStackTrace();
            }
    }
}
```

Figura 5 Creación del cliente Multicast

Por último, el método main(), mostrado en la figura 6, contiene la validación para recibir el nombre de usuario en la ejecución del programa y un ciclo infinito donde se despliega la información referente a los mensajes que se enviaran al grupo multicast.

```
public static void main(String[] args) throws Exception {
    if (args.length < 1){
        System.out.println("Use: java Chat <username>");
        return;
    }
    new Worker().start(); //Starts server
    String name = args[0];
    Scanner input = new Scanner(System.in,"Cp850");
    System.out.println(name + " te has unido al chat");
    StringBuilder message = new StringBuilder(); //StringBuilder is used to avoid String object duplication
    System.out.println("Ingrese el mensaje a enviar:");
    while(true) {
        message.append(name + " dice " + input.nextLine()); //Recieves keyboard input
        sendMessage(
            message.toString().getBytes(), ip, 50000);
        message.delete(0, message.length());
    }
}
```

Figura 6 Método principal de la clase Chat

Funcionamiento

Creación de la máquina virtual

Para esta práctica se hace uso de una máquina virtual Windows Server 2012, creada en Azure, a continuación, se explica a detalle la creación de dicha máquina virtual.

Creación paso a paso:

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" como se muestra en la figura 7.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with links like 'Inicio - Microsoft Azure', 'Buscar recursos, servicios y documentos (G+)', and user information ('kramirezg1400@alumn... INSTITUTO POLITECNICO NACION...'). Below the navigation bar is the main dashboard titled 'Servicios de Azure' (Azure Services). On the left, there's a sidebar with sections for 'Crear un recurso' (Create a resource), 'Suscripciones' (Subscriptions), 'Microsoft Learn' (with 101 courses), and 'Vínculos útiles' (useful links). The central area is focused on 'Máquinas virtuales' (Virtual Machines), which is highlighted with a yellow box. Other service tiles include 'Cuentas de almacenamiento' (Storage accounts), 'SQL Database', 'Azure Cosmos DB', 'Servicios de Kubernetes', 'Todos los servicios' (All services), 'Todos los recursos' (All resources), 'Panel' (Dashboard), 'Microsoft Defender for Cloud', 'Administración de costos' (Cost management), and 'Aplicación móvil de Azure' (Azure mobile app). At the bottom, there's a taskbar with icons for various Windows apps and a weather forecast for Madrid (23°C, Lluvia ligera).

Figura 7 Selección de “Máquinas virtuales” dentro de la cuenta de Microsoft Azure

3. En la figura 8 se observa que aún no se tiene una máquina virtual, se creara una nueva dando click en el botón “+Crear”.
4. Seleccionar la opción de “Máquina virtual”

Inicio >

Máquinas virtuales

Instituto Politecnico Nacional (correo.ipn.mx)

+ Crear Cambiar al modo clásico Reservas Administrar vista Actualizar Exportar a CSV Abrir consulta Asignar etiquetas Iniciar Reiniciar ...

+ Máquina virtual Comenzar con una configuración preestablecida

Mostrando de 0 a 0 de 0 registros.

Nombre ↑↓ Suscripción ↑↓ Grupo de recursos ↑↓ Ubicación ↑↓ Estado ↑↓ Sistema operativo ↑↓ Tamaño ↑↓ Dirección IP públ..

No hay máquinas virtuales para mostrar

Crea una máquina virtual que ejecuta Linux o Windows. Seleccione una imagen de Marketplace o use una imagen personalizada propia.

Más información acerca de Windows Virtual Machines
Más información sobre Linux Virtual Machines

Figura 8 Selección de "Crear" nueva máquina virtual

5. Se crea un grupo de recursos, en este caso llamado “Tarea4Equipo12”
6. El nombre de la máquina virtual es “Tarea-4-12”
7. Seleccionar la región donde se creará la máquina virtual
8. Seleccionar la imagen, en este caso se elige Windows Server 2012 R2 Datacenter – Gen1.
9. Dar click en "Seleccionar tamaño" de la máquina virtual, en este caso se selecciona Standard_B2s – 2vcpu, 4 GiB.
10. En tipo de autenticación seleccionamos "Contraseña".
11. Ingresamos el nombre del usuario, en este caso: “Equipo12”

12. Ingresamos la contraseña y confirmamos la contraseña. La contraseña debe tener al menos 12 caracteres, debe al menos una letra minúscula, una letra mayúscula, un dígito y un carácter especial.

13. En las "Reglas de puerto de entrada" se deberá dejar abierto el puerto de entrada RDP (3389).

Las configuraciones del paso 5 al 13 se muestran en la figura 9.

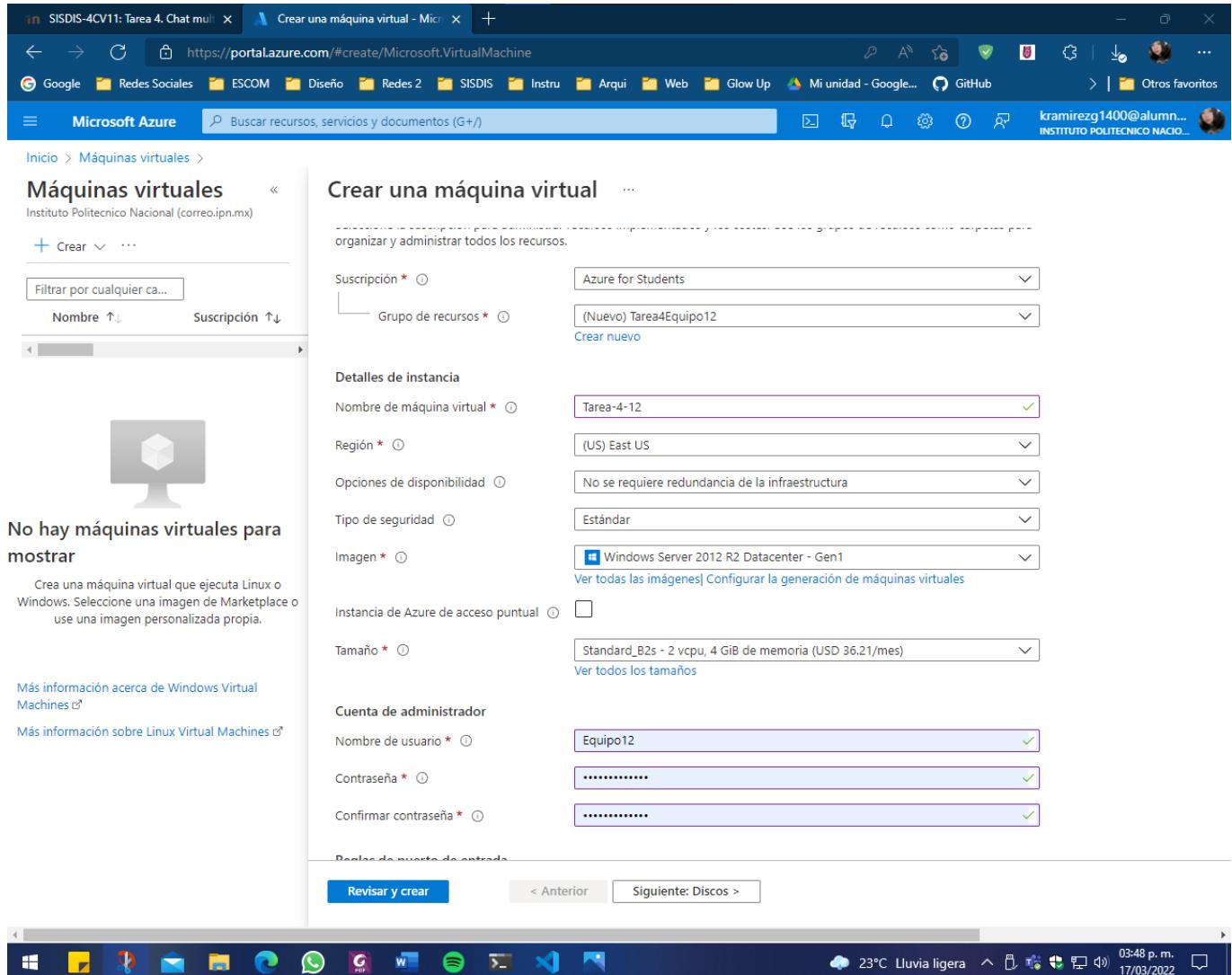


Figura 9 Datos básicos para la creación de la máquina virtual

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

15. Seleccionar el tipo de disco de sistema operativo, en este caso se ocupa HDD estándar. (véase Figura 10)

NOTA: los demás campos se dejan con los valores por defecto

Máquinas virtuales

Crear una máquina virtual

Datos básicos Discos Redes Administración Opciones avanzadas Etiquetas Revisar y crear

Las máquinas virtuales de Azure tienen un disco de sistema operativo y un disco temporal para el almacenamiento a corto plazo. Puede asociar discos de datos adicionales. El tamaño de la máquina virtual determina el tipo de almacenamiento que puede usar y la cantidad de datos que permiten los discos. [Más información](#)

Opciones de disco

Tipo de disco del sistema operativo *

El tamaño de la máquina virtual seleccionada es compatible con los discos premium. Se recomienda SSD Premium para elevadas cargas de trabajo de E/S por segundo. Las máquinas virtuales con discos SSD Premium optan al acuerdo de nivel de servicio de conectividad del 99,9%.

Eliminar con VM

Cifrado en el host

El cifrado en el host no está registrado para la suscripción seleccionada. [Más información sobre cómo habilitar esta característica](#)

Tipo de cifrado *

Habilitar compatibilidad con Ultra Disks

El disco Ultra se admite en las zonas de disponibilidad 1,2,3 para el tamaño de VM seleccionado (Standard_B2s).

Discos de datos para Tarea-4-11

Puede agregar y configurar discos de datos adicionales para su máquina virtual o asociar discos existentes. Esta máquina virtual también incluye un disco temporal.

ID UN	Nombre	Tamaño	Tipo de disco	Almacena	Eliminar con VM

Revisar y crear < Anterior Siguiente: Redes >

Figura 10 Configuración de "Discos" de la máquina virtual

16. Dar click en el botón "Siguiente: Redes>" (véase la Figura 11)

En esta pestaña no es necesario cambiar nada, se dejan los valores por defecto.

Máquinas virtuales

Crear una máquina virtual

Datos básicos Discos Redes Administración Opciones avanzadas Etiquetas Revisar y crear

Configure la tarjeta de interfaz de red (NIC) a fin de definir la conectividad de red para la máquina virtual. Puede controlar los puertos y la conectividad entrante y saliente con reglas de grupos de seguridad o bien aplicar una solución de equilibrio de carga ya existente. [Más información](#)

Interfaz de red

Al crear una máquina virtual, se crea una interfaz de red automáticamente.

Red virtual * (nuevo) Tarea4Equipo12-vnet
Crear nuevo

Subred * (nuevo) default (10.0.0.0/24)

IP pública (nuevo) Tarea4-12-ip
Crear nuevo

Grupo de seguridad de red de NIC Ninguno Básico Opciones avanzadas

Puertos de entrada públicos * Ninguno Permitir los puertos seleccionados

Seleccionar puertos de entrada * RDP (3389)

Advertencia: Esto permitirá que todas las direcciones IP accedan a la máquina virtual. Esto solo se recomienda para las pruebas. Use los controles avanzados de la pestaña Redes a fin de crear reglas para limitar el tráfico entrante a las direcciones IP conocidas.

Revisar y crear < Anterior Siguiente: Administración >

Figura 11 Configuración de "Redes" de la máquina virtual

17. Dar click en el botón "Siguiente: Administración>"
18. En el campo "Diagnóstico de arranque" seleccionar "Desactivado". (véase la figura 12).

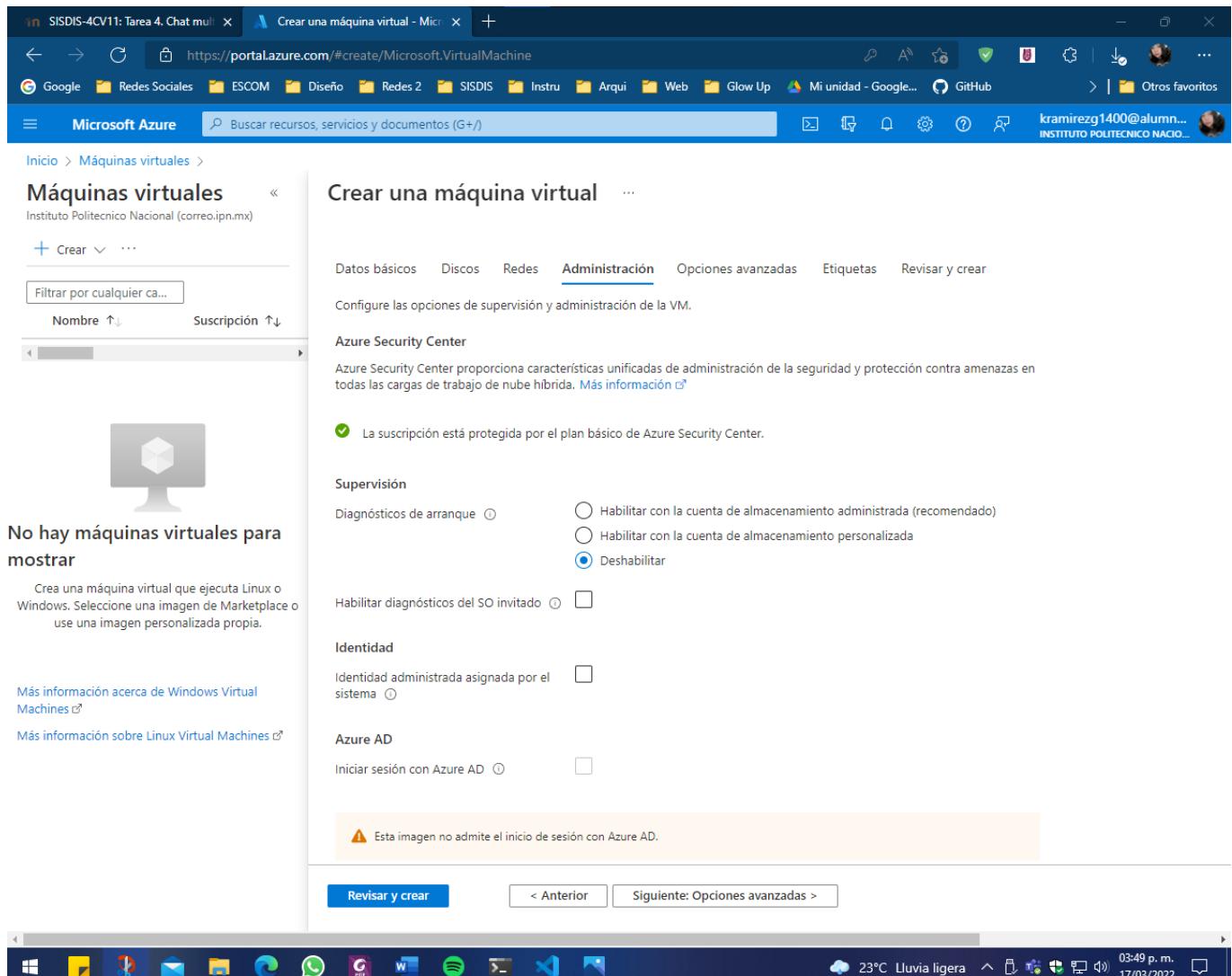


Figura 12 Pestaña “Administración” de la máquina virtual

En la pestaña de “opciones avanzadas” todos los valores se quedan por defecto.

19. Dar click en el botón "Revisar y crear". (véase la Figura 13)

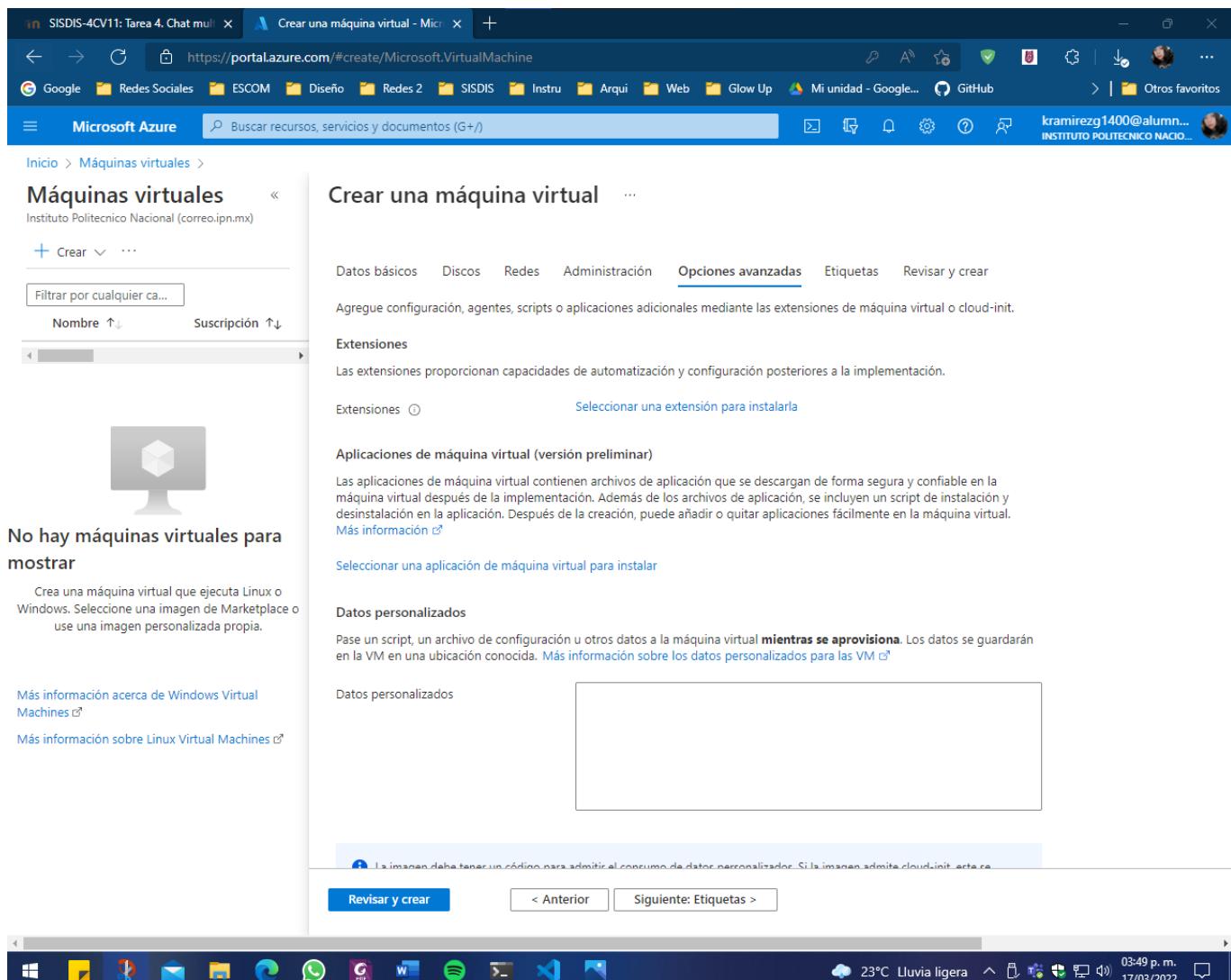


Figura 13 Configuración de "opciones avanzadas" de la máquina virtual

20. Dar click en el botón "Crear". (véase la Figura 14)

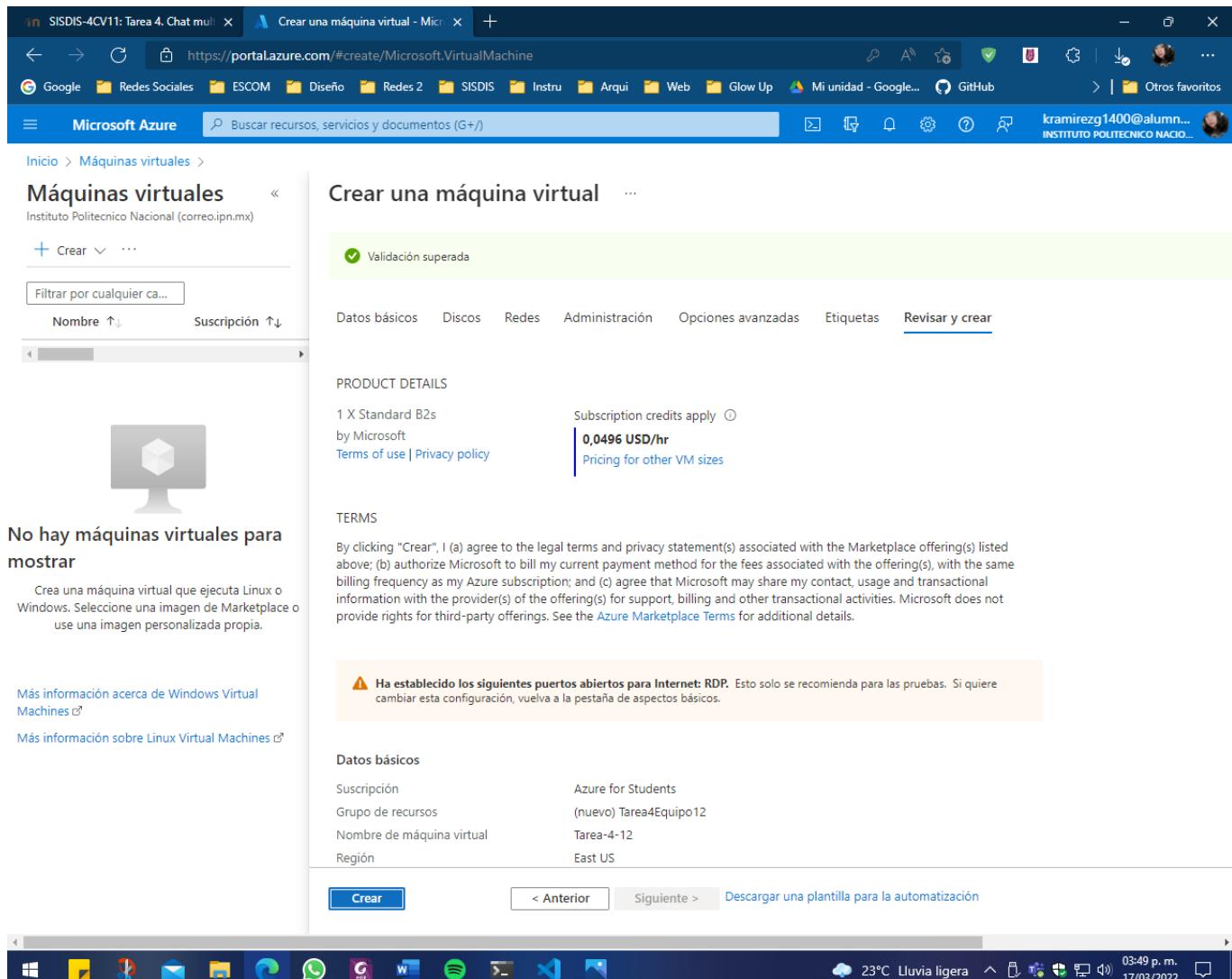


Figura 14 Opción “Crear” máquina virtual

21. Dar click a la campana de notificaciones (barra superior de la pantalla) para verificar que la máquina virtual se haya creado. (véase la figura 15)

The screenshot shows a Microsoft Azure portal window. At the top, there are two tabs: 'SISDIS-4CV11: Tarea 4. Chat mult...' and 'CreateVm-MicrosoftWindowsServer...'. The address bar shows the URL: <https://portal.azure.com/#blade/HubsExtension/DeploymentDetailsBlade/overview/id/%2Fsubscriptions%2Ff...>. Below the address bar, there's a navigation bar with links like 'Google', 'Redes Sociales', 'ESCOM', 'Diseño', 'Redes 2', 'SISDIS', 'Instru', 'Arqui', 'Web', 'Glow Up', 'Mi unidad - Google...', 'GitHub', and 'Otros favoritos'. On the right side of the header, it shows the user's name 'kramirezg1400@alumn...' and 'INSTITUTO POLITÉCNICO NACIONAL'.

The main content area displays the deployment details for 'CreateVm-MicrosoftWindowsServer.WindowsServer-201-20220317154750'. It shows a green checkmark icon and the message 'Se completó la implementación' (Implementation completed). Below this, it lists the implementation details: Nombre de implementación: CreateVm-MicrosoftWindowsServer..., Suscripción: Azure for Students, Grupo de recursos: Tarea4Equipo12. It also shows the start time: Hora de inicio: 17/3/2022, 15:50:28 and the correlation ID: Id. de correlación: 73cf5e5d-1642-49b7-ad0c-6c5ba3... There are two buttons at the bottom: 'Ir al recurso' (Go to resource) and 'Crear otra VM' (Create another VM).

On the right side of the page, there are several promotional cards:

- Implementación correcta**: La implementación "CreateVm-MicrosoftWindowsServer.WindowsServer-201-20220317154750" se realizó correctamente en el grupo de recursos "Tarea4Equipo12".
- Cost Management**: Obtenga una notificación para dentro del presupuesto y evíe inesperados en su factura. [Configurar alertas de costo >](#)
- Microsoft Defender for Cloud**: Proteja sus aplicaciones e información. [Ir a Microsoft Defender for C...](#)
- Tutoriales gratuitos de Microsoft**: Comience a aprender hoy >
- Trabajar con un experto**: Los expertos de Azure son los proveedores de servicios que administran sus recursos en primera línea de soporte técnico. [Buscar un experto de Azure >](#)

The taskbar at the bottom of the screen shows various pinned icons, including File Explorer, Microsoft Edge, and other Microsoft applications. The system tray indicates the date as 17/03/2022, the time as 03:51 p.m., and the weather as 23°C Lluvia ligera.

Figura 15 Creación exitosa de la máquina virtual

22. Dar click en el botón "Ir al recurso". En la página de puede ver la dirección IP pública de la máquina virtual. Esta dirección puede cambiar cada vez que se apague y se encienda la máquina virtual. (véase la figura 16)

Tarea-4-12

Máquina virtual

Buscar (Ctrl+ /)

Conectar Iniciar Reiniciar Detener Captura Eliminar Actualizar Abrir en dispositivos móviles CLI / PS ...

Información general

Estado: En ejecución

Ubicación: East US

Sistema operativo: Windows

Tamaño: Standard B2s (2 vcpu, 4 GiB de memoria)

Dirección IP pública: 20.25.39.240

Red virtual/subred: Tarea4Equipo12-vnet/default

Nombre DNS: Sin configurar

Etiquetas: Haga clic aquí para agregar etiquetas.

Propiedades Supervisión Funcionalidades (7) Recomendaciones Tutoriales

Máquina virtual		Redes	
Nombre del equipo:	Tarea-4-12	Dirección IP pública:	20.25.39.240
Estado de mantenimiento:	-	Dirección IP pública (IPv6):	-
Sistema operativo:	Windows	Dirección IP privada:	10.0.0.4
Publicador:	MicrosoftWindowsServer	Dirección IP privada (IPv6):	-
Oferta:	WindowsServer	Red virtual/subred:	Tarea4Equipo12-vnet/default
Plan:	2012-R2-Datcenter	Nombre DNS:	Configurar
Generación de VM:	V1		
Estado del agente:	Not Ready		
Versión del agente:	Unknown		

Tamaño: Standard B2s

Figura 16 Información general de la máquina virtual

Configuración de la máquina virtual

Para conectarse remotamente a la máquina virtual es necesario ir al menú en la opción “conectar” y dar click en “RDP” como se muestra en la figura 17.

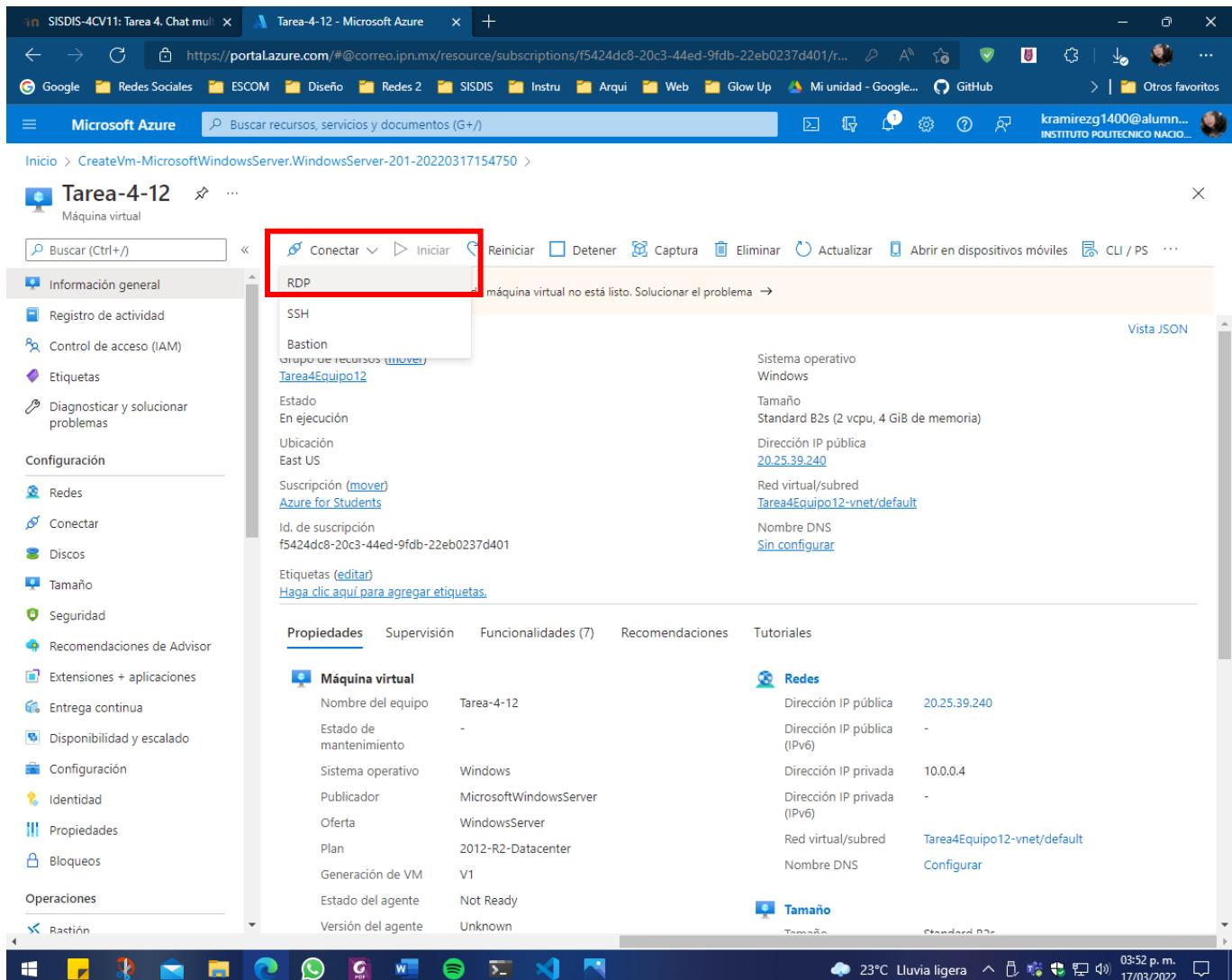


Figura 17 Configuración de una máquina virtual con Windows Server en Azure

Una vez seleccionada la opción de “RDP” aparece la pantalla mostrada en la figura 18.

Se tiene que dar click en el botón que dice “Descargar Archivo RDP” el cual servirá para establecer la conexión remota con la máquina virtual que se creó anteriormente. Al hacerlo se descargará un archivo con la misma extensión y nombre de la máquina virtual, en este caso “Tarea-4-12”.

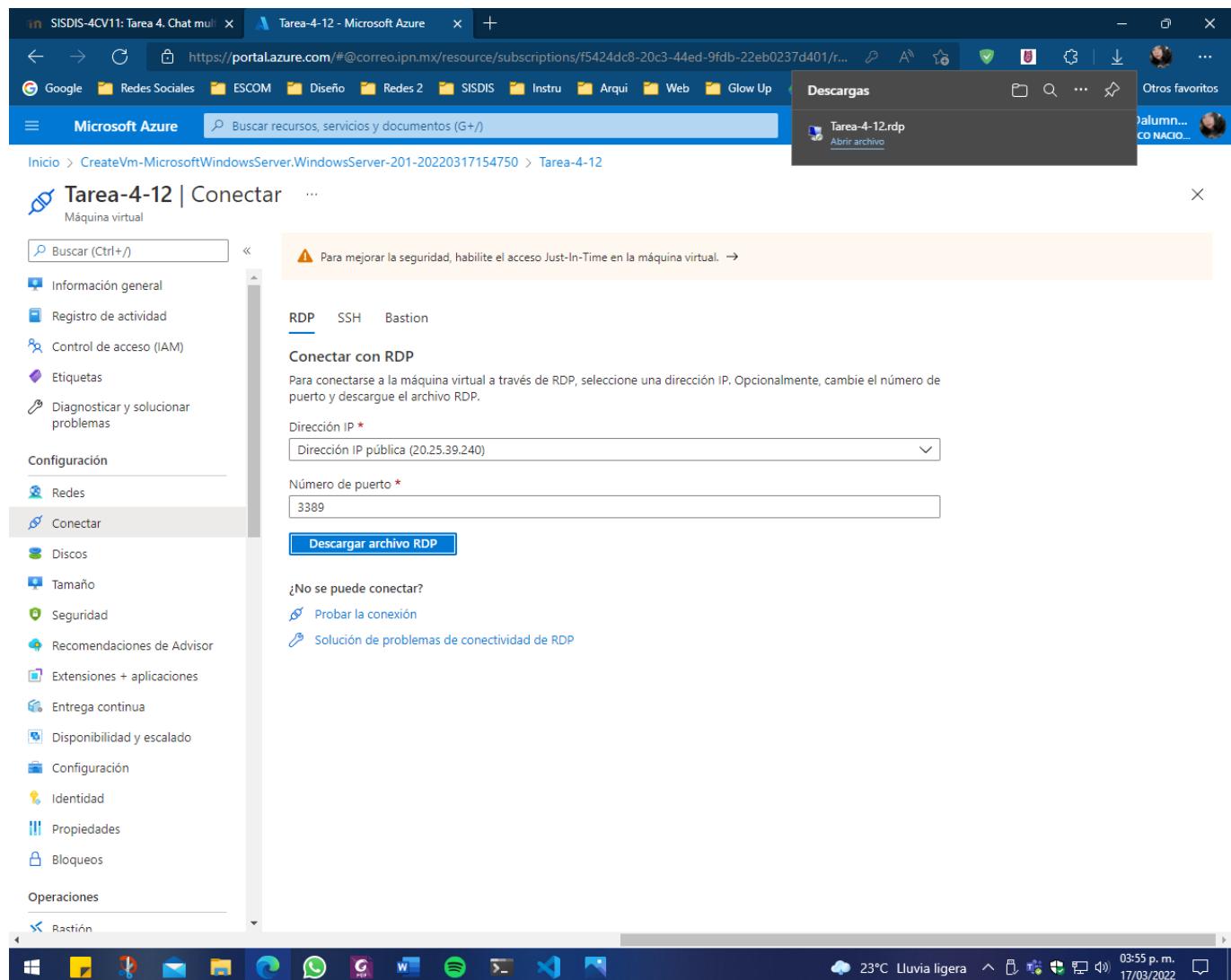


Figura 18 Descarga del archivo RDP

Una vez descargado el archivo RDP abrimos su ubicación y con click derecho se debe seleccionar la opción “Editar”. (véase la figura 19).

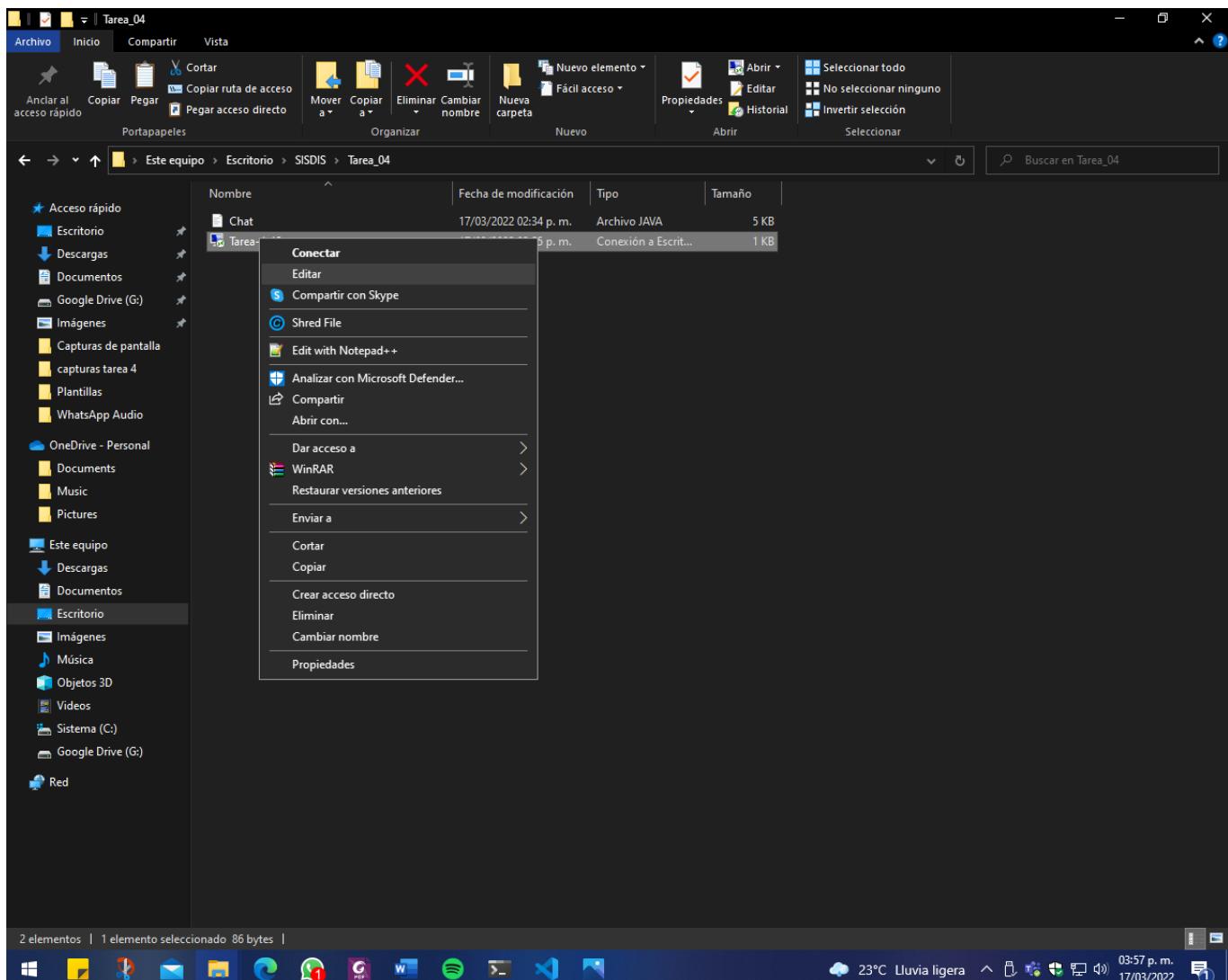


Figura 19 Configuración de RDP parte 1

Se abrirá una ventana llamada “Conexión a Escritorio Remoto”, en el menú se debe seleccionar la opción “Recursos locales”, dentro de esa pestaña seleccionar el botón que dice “Más...”, se abre una nueva pestaña, hay que desplegar el submenú “Unidades” y se selecciona la unidad que se va a compartir con la máquina virtual, para este caso se usó una USB la cual esta identificada con la letra “D:” y cuyo nombre es “COMPARTIDO”. Al finalizar se selecciona la opción “Aceptar”. (véase la figura 20).

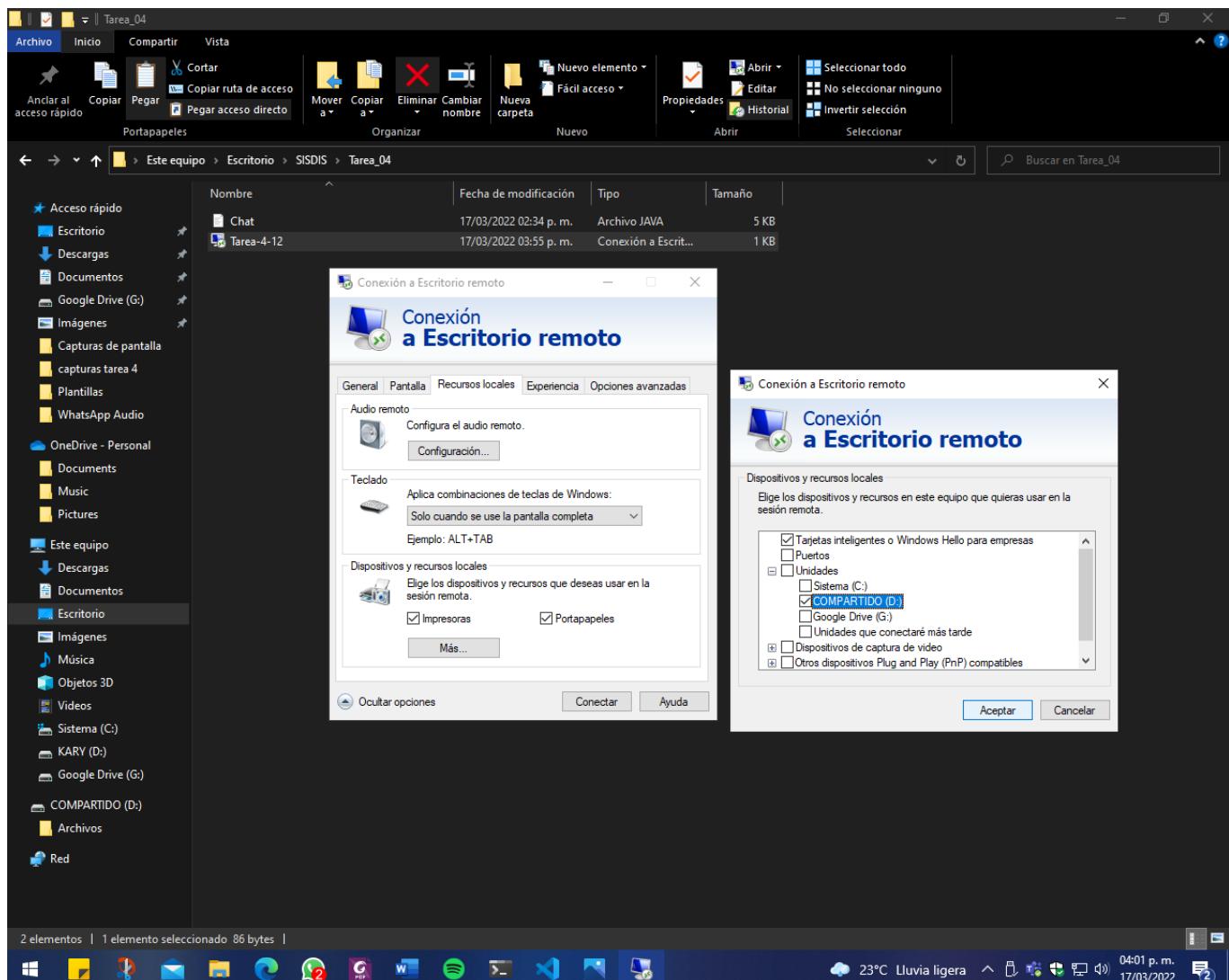


Figura 20 Configuración RDP parte 2

Para poder ejecutar el Chat multicast en la máquina virtual es necesario compartir información, en este caso, el instalador de java JDK 15 y el programa Chat.java. Para ello en la ventana de “Conexión a escritorio remoto” hay que dar click en el botón que dice “Conectar” y aparecerá el mensaje mostrado en la figura 21.

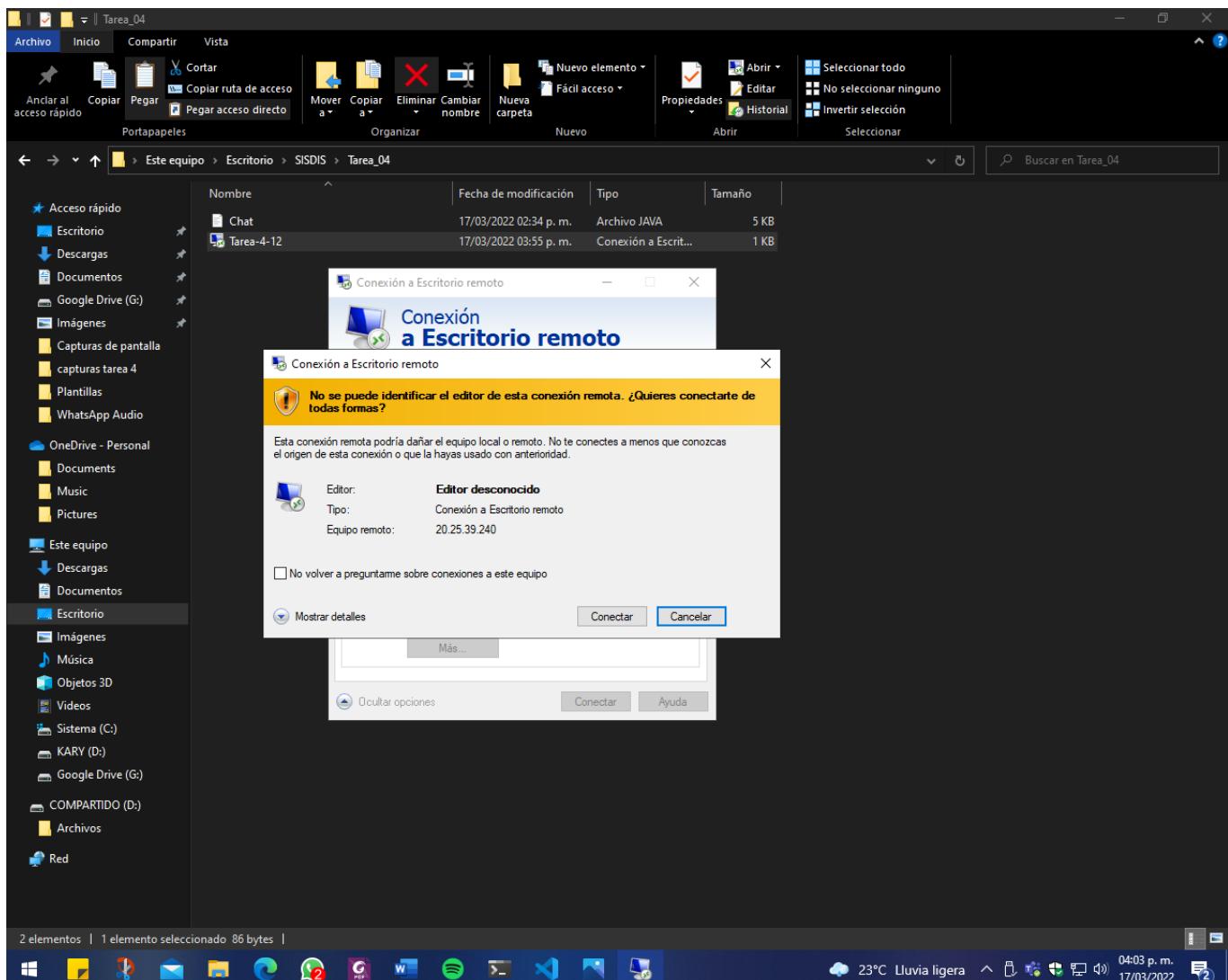


Figura 21 Conexión RDP parte 1

En la ventana emergente seleccionar la opción “Conectar” y se desplegará la ventana de la figura 22 donde hay que llenar con los datos (nombre de usuario y contraseña) establecidos en la creación de la máquina virtual en Azure.

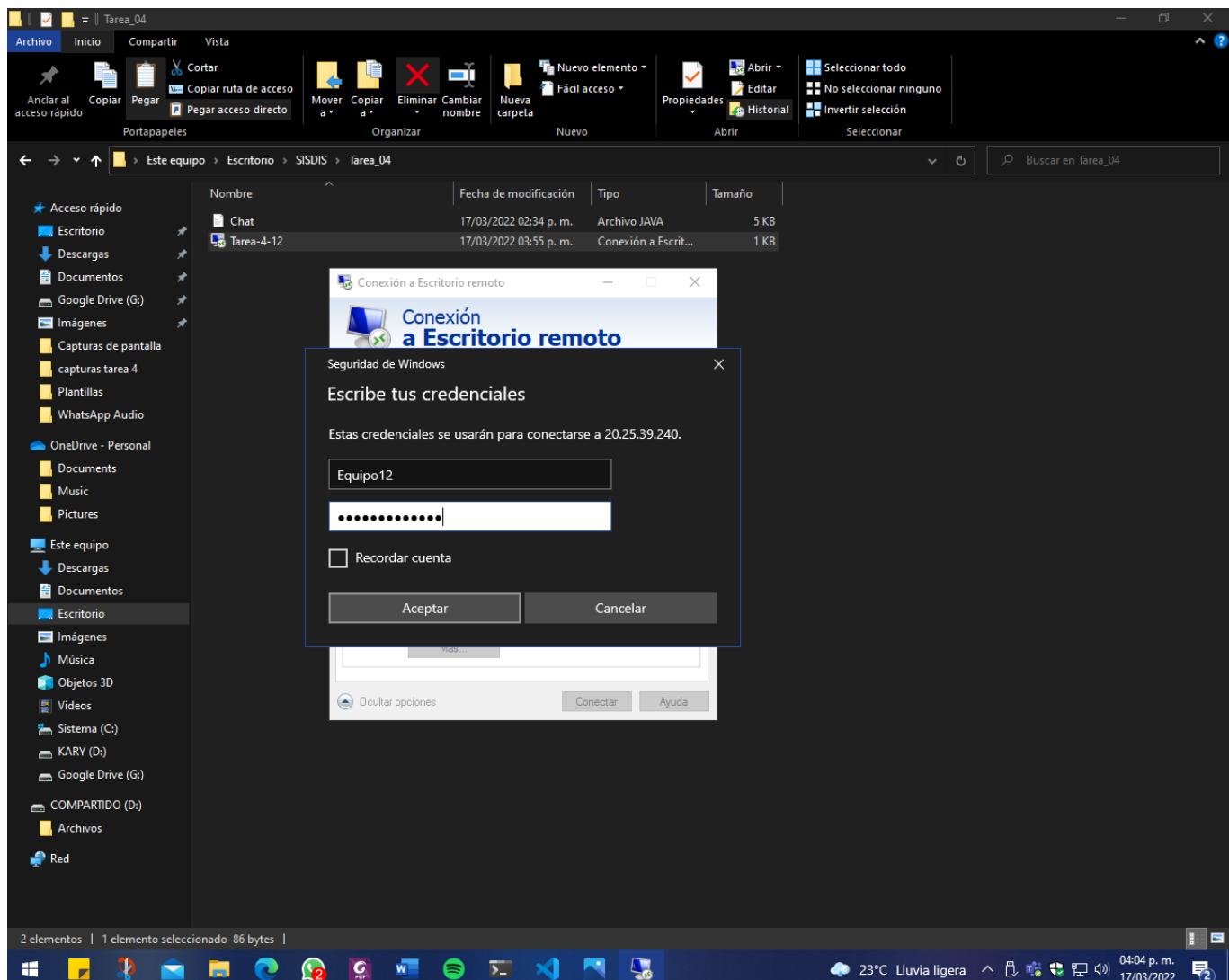


Figura 22 Conexión RDP parte 2

Nuevamente aparece una ventana emergente indicando la configuración de autenticación de la máquina virtual, como solo el Equipo 12 tiene acceso a este recurso, se procede a dar click en el botón “si”. (véase la figura 23).

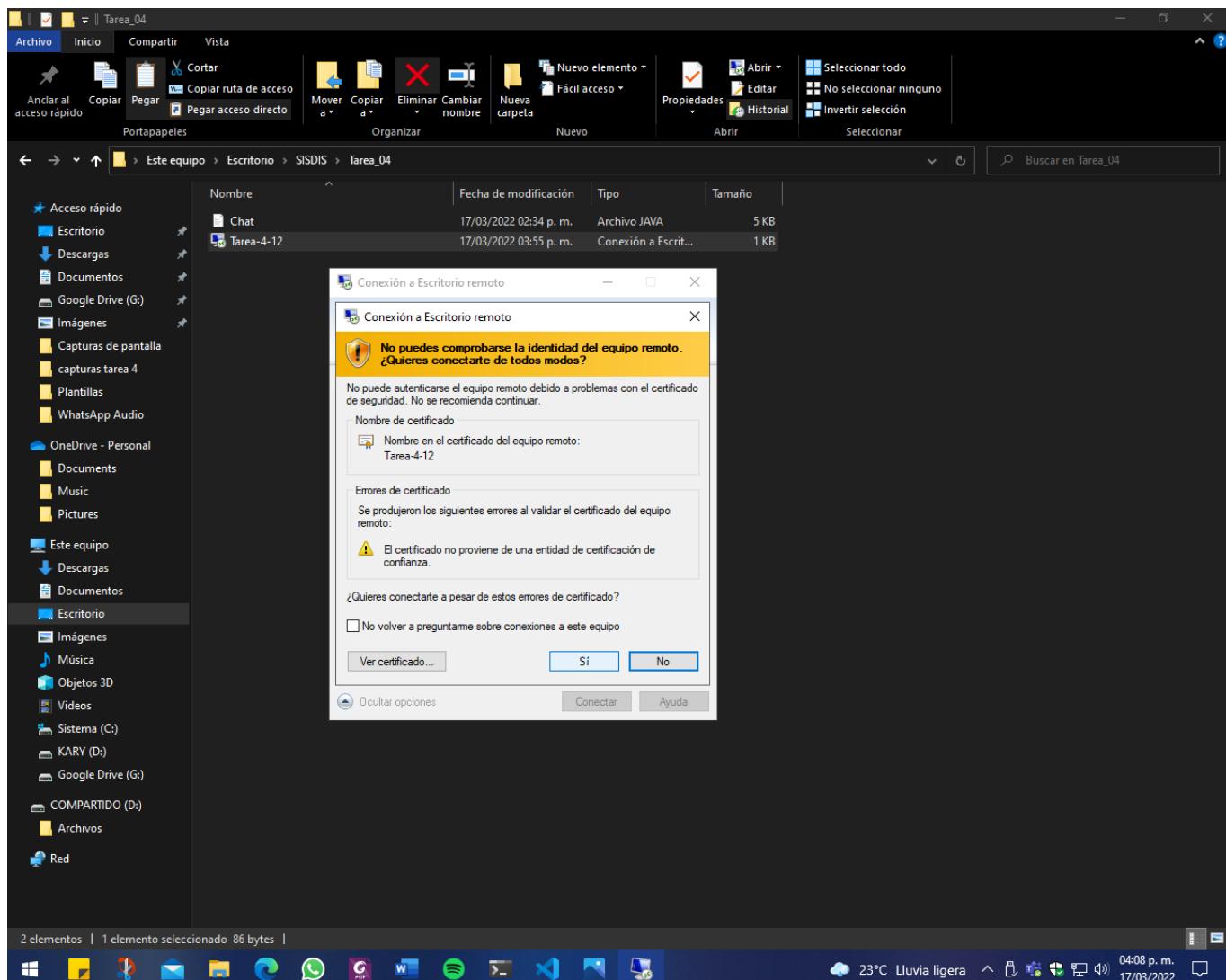


Figura 23 Conexión RDP parte 3

Una vez hecha la acción anterior se abre la ventana de la Figura 24 la cual muestra la conexión remota a la máquina virtual de Azure con Windows Server 2012.

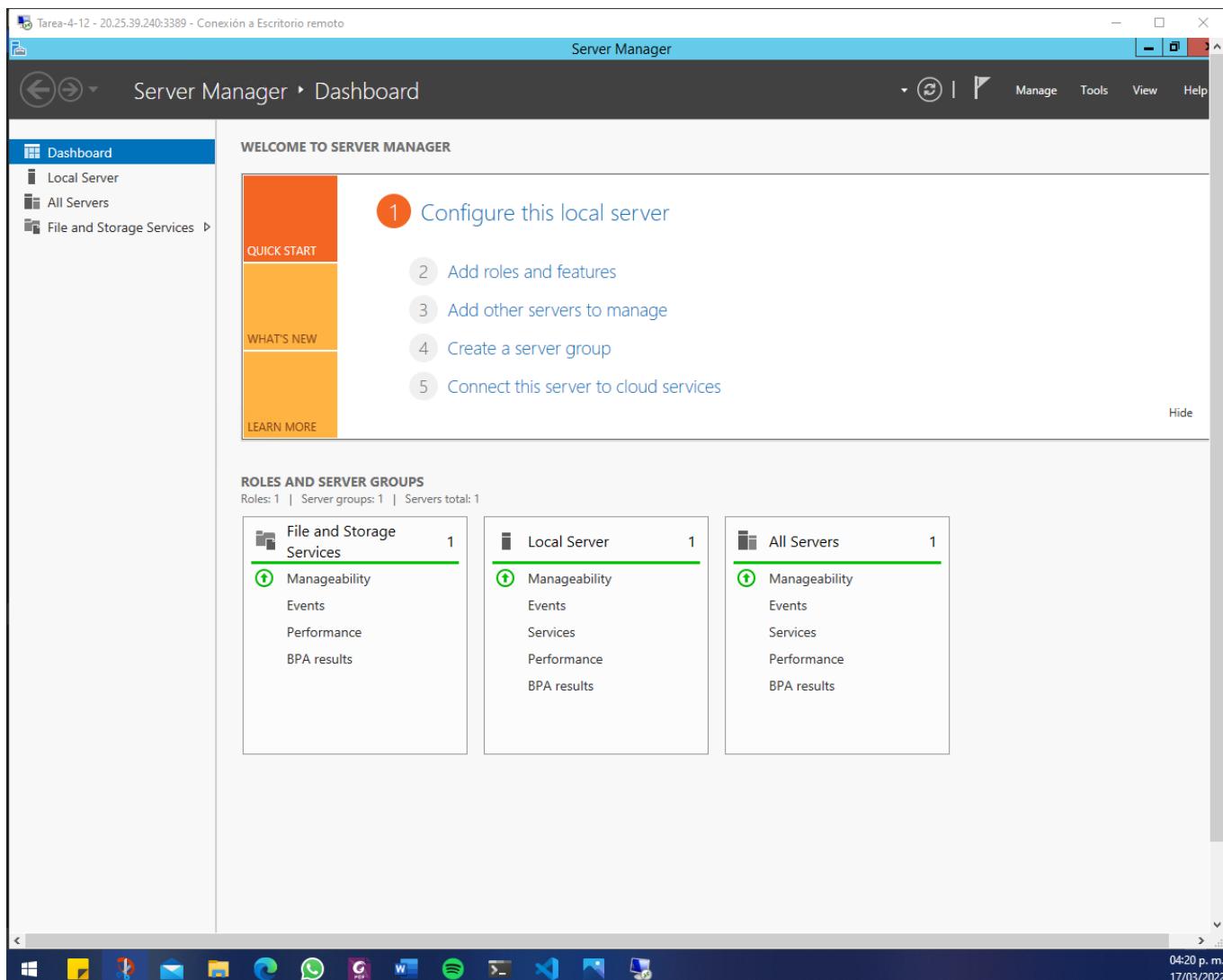


Figura 24 Inicio de la máquina virtual de Azure con Windows Server

Obsérvese que la maquina tiene como nombre “Tarea4-8”.

Abriendo un explorador de archivos se puede apreciar cómo se encuentra habilitada nuestra unidad compartida desde la máquina local a la máquina virtual de Azure, en este caso aparece como “D on DESKTOP-I4R5UOI” haciendo referencia a que se compartió la unidad con letra “D:” desde la computadora local que tiene por nombre “DESKTOP-I4R5UOI”. (véase la Figura 25).

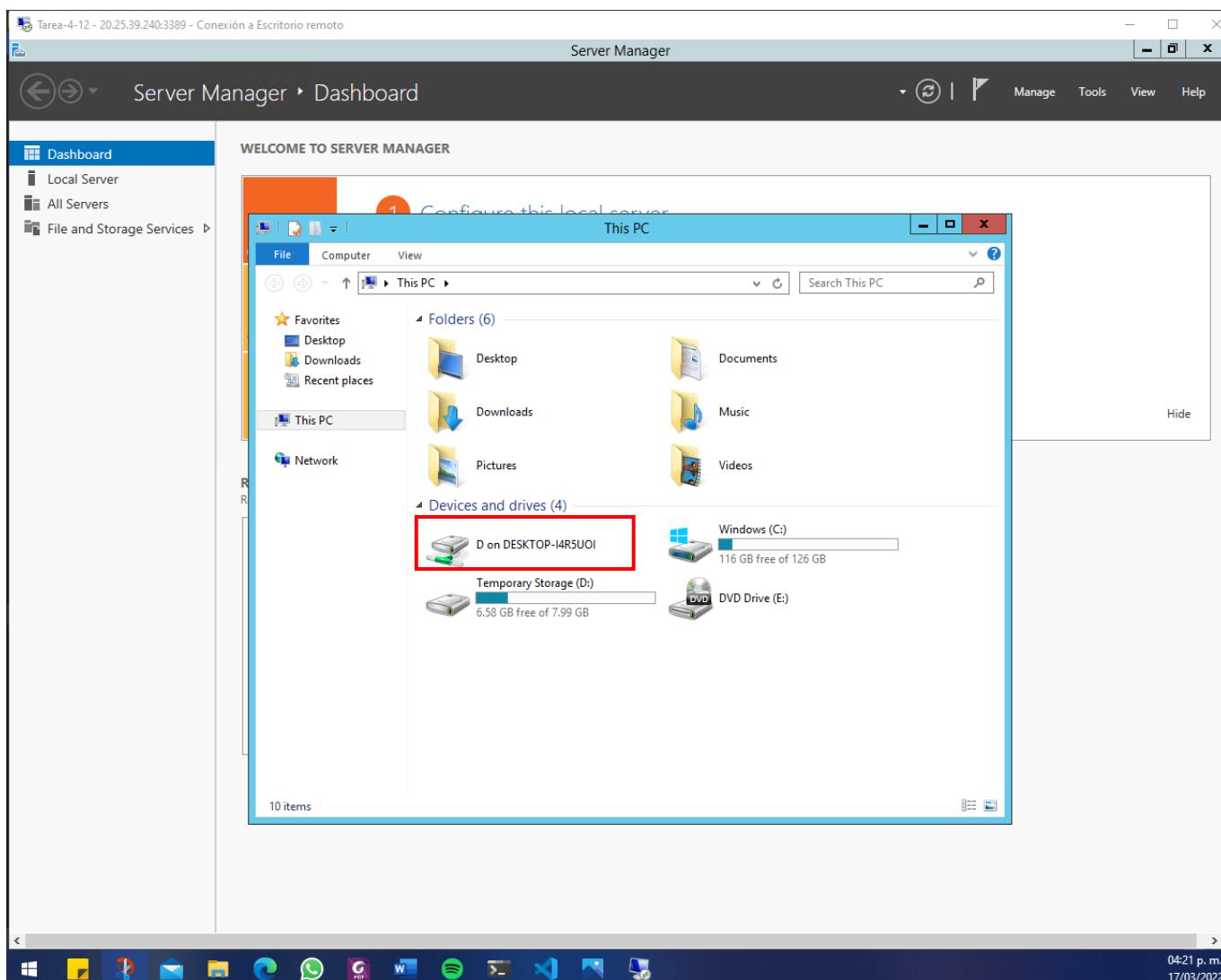


Figura 25 Nombre de la unidad compartida con la máquina virtual

Dando click en el dispositivo compartido, se deben tener los archivos que se compartirán en la máquina virtual, se deben copiar:

- jdk_15.0.2_windows-x64_bin
- Chat.java

La copia de archivos se muestra en la Figura 26.

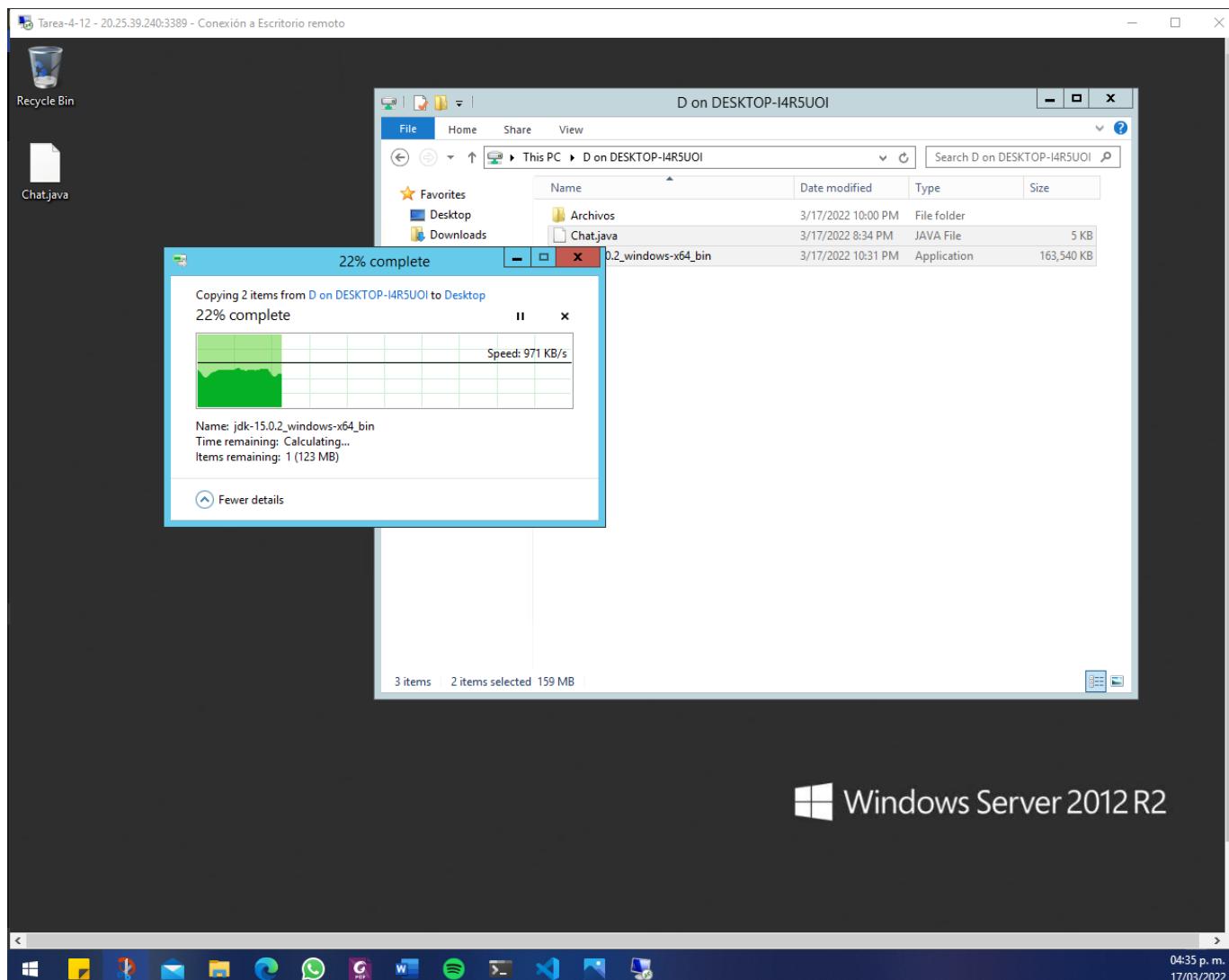


Figura 26 Copiado de archivos desde la unidad compartida a la máquina virtual

Después hay que instalar el JDK 15 en la máquina virtual. Para ello el archivo `jdk_15.0.2_windows-x64_bin` se ejecuta como administrador y se hace la instalación de manera genérica. (véase la Figura 27).

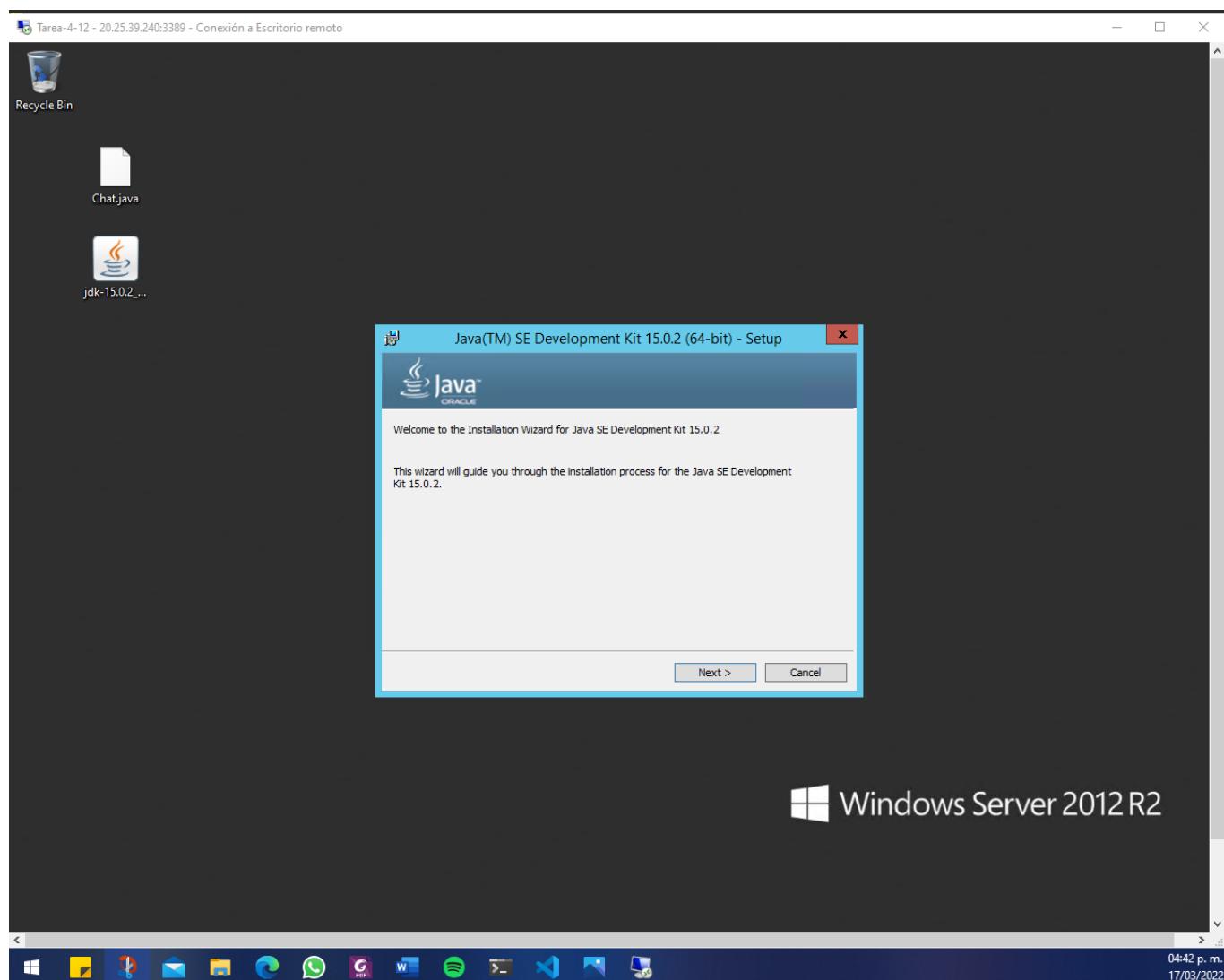


Figura 27 Instalación del JDK 15 en la máquina virtual

Para confirmar que se realizó la instalación de manera exitosa, se abrió una terminal para checar la versión de java como se muestra en la Figura 28.

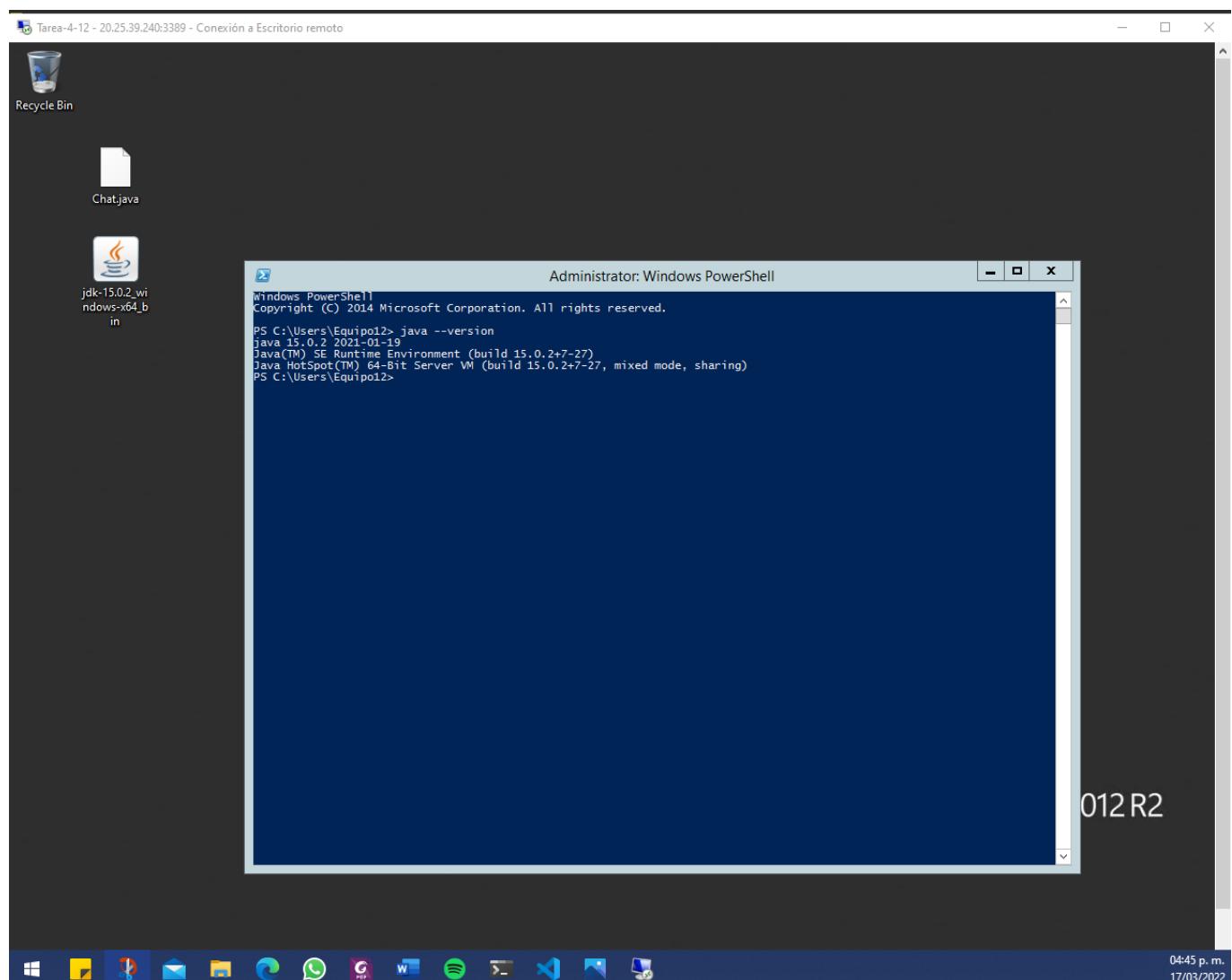


Figura 28 Verificación de la versión instalada de Java

Compilación y ejecución del programa

Para compilar y ejecutar el programa Chat.java se utilizan tres terminales en la máquina virtual. (véase la Figura 29).

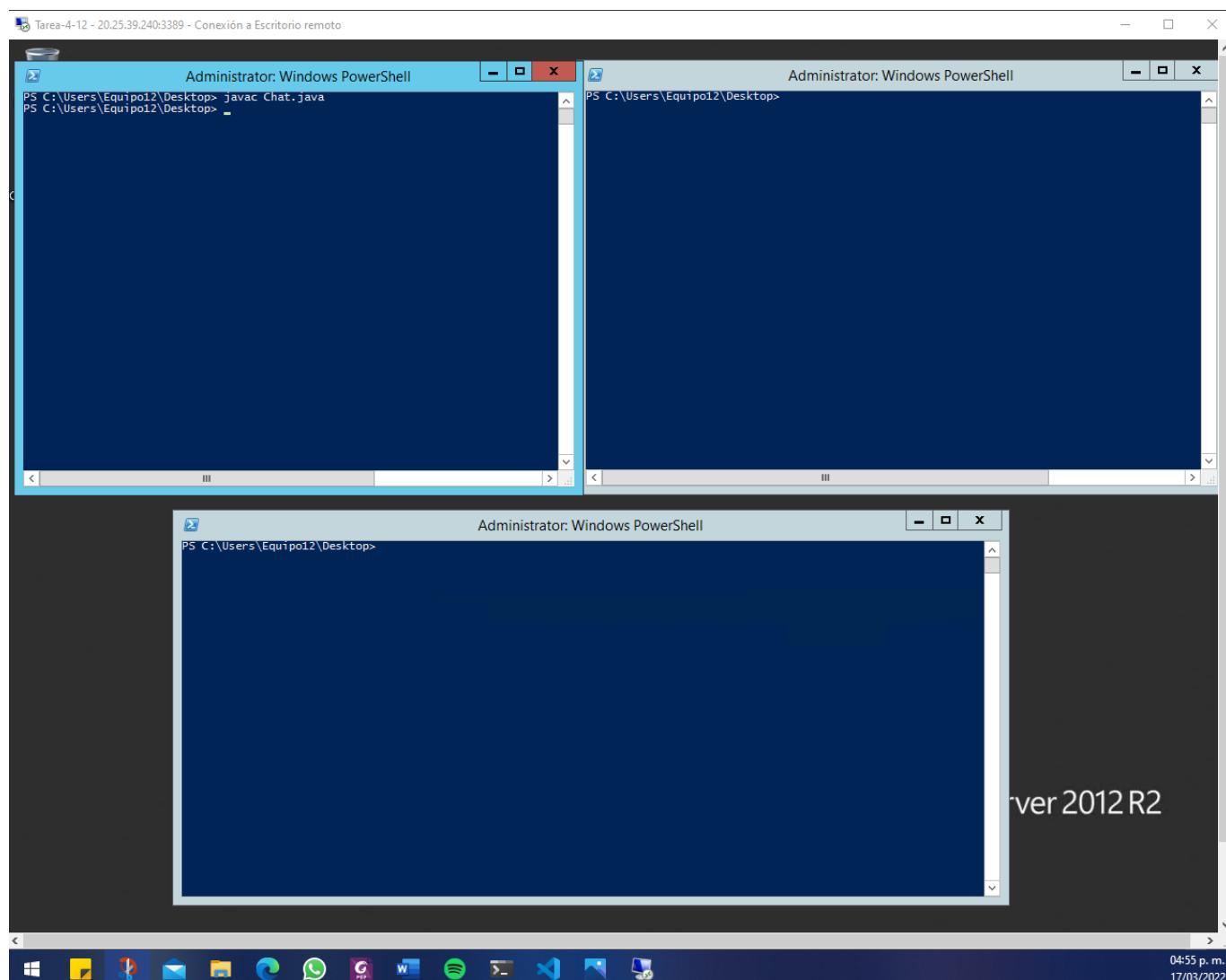


Figura 29 Compilación del programa Chat.java en la máquina virtual

Para la ejecución del programa se utilizaron los usuarios solicitados: Pedro, Juan y Ana. (véase la Figura 30).

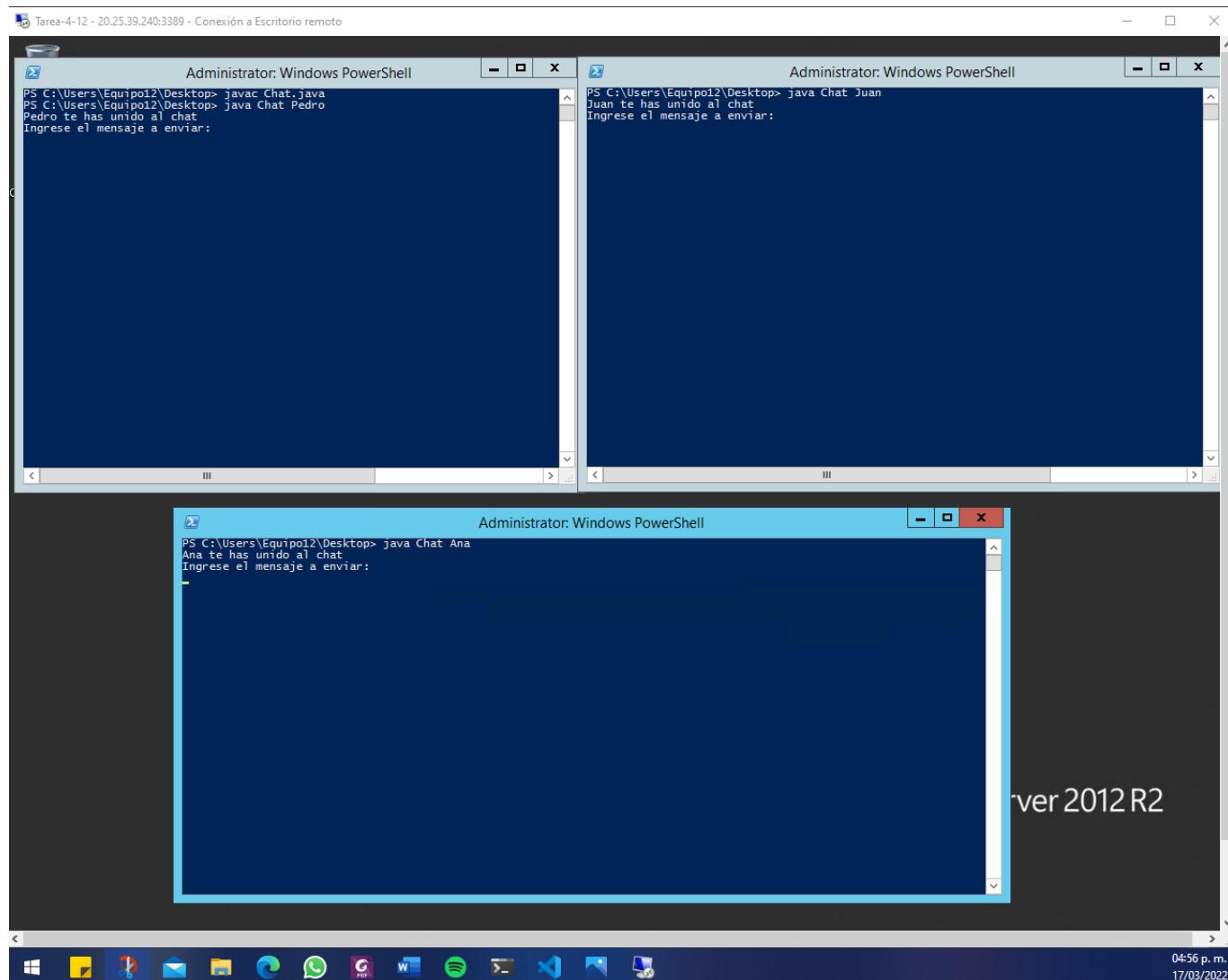


Figura 30 Ejecución del chat multicast con 3 usuarios: Pedro, Juan y Ana

Para una mejor apreciación de la conversación solicitada, se añade un usuario de prueba con el nombre “ChatFinal” cuya finalidad es solo estar a la escucha de mensajes como se muestra en la Figura 31.

También se puede apreciar el paso de mensajes dentro del grupo multicast siguiendo la conversación entre Pedro, Juan y Ana, como se solicita en los requerimientos de la práctica.

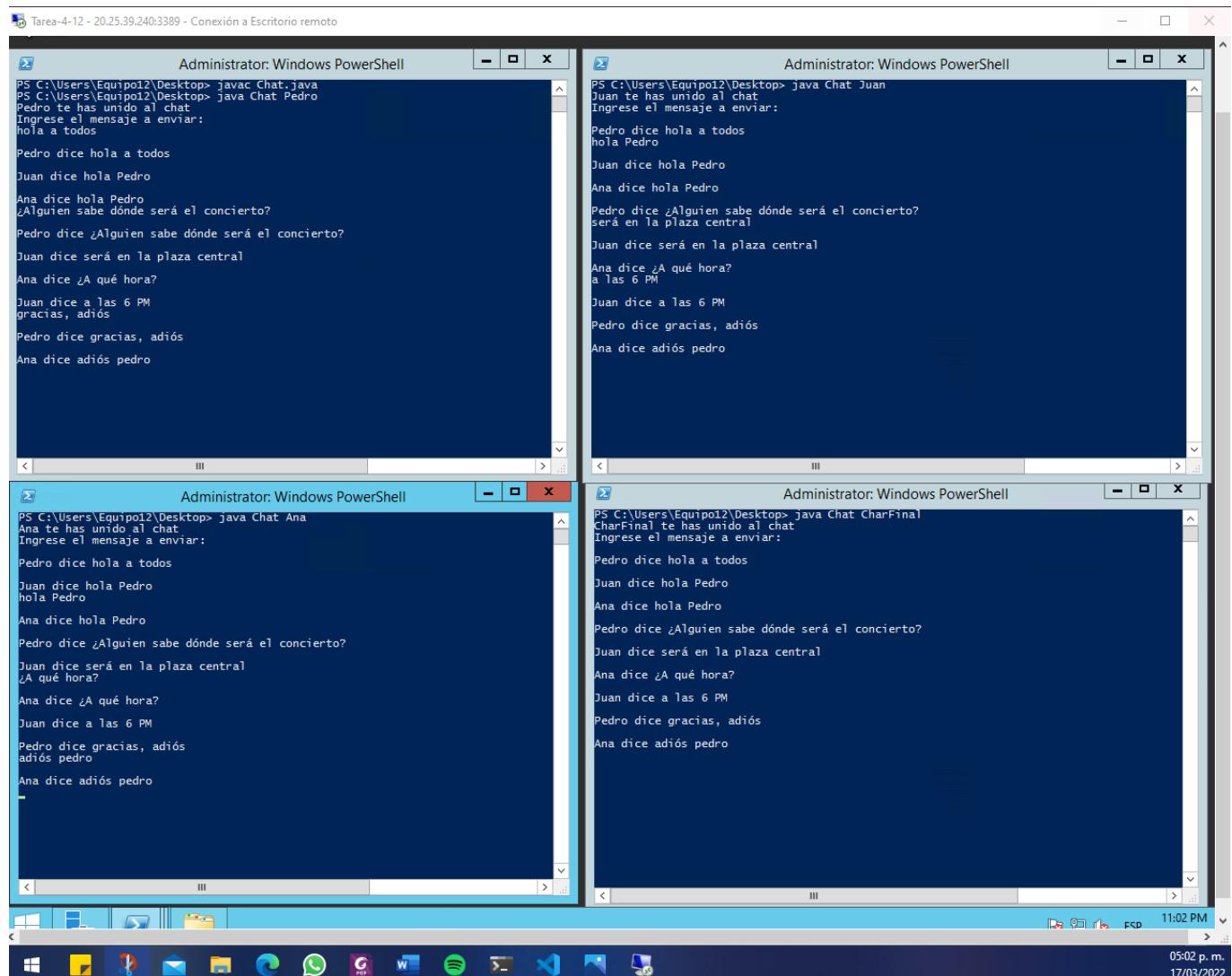


Figura 31 Usuario de prueba "ChatFinal"

La vista de la conversación de cada usuario se muestra en las Figuras 32,33,34 y 35.

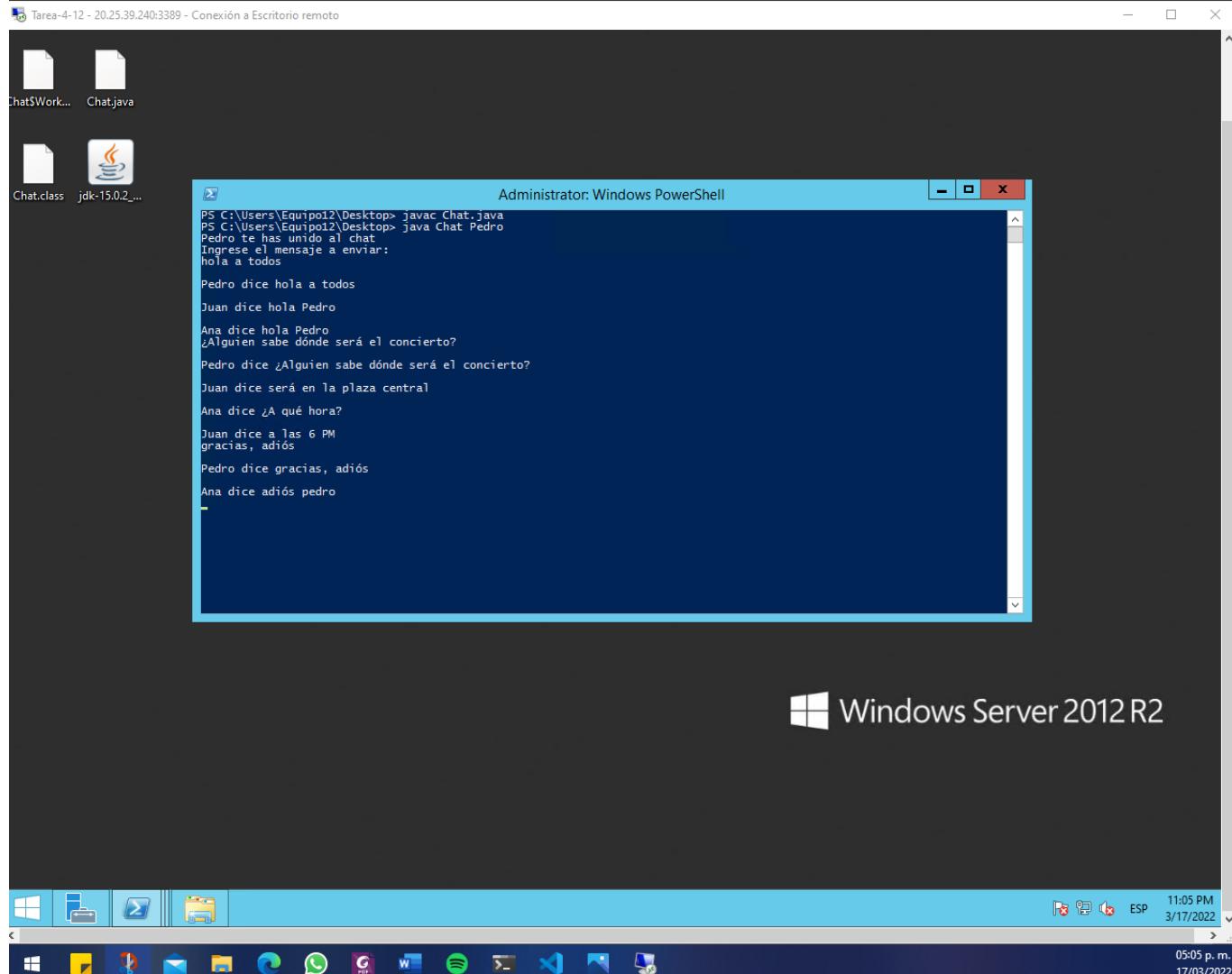


Figura 32 Chat de Pedro

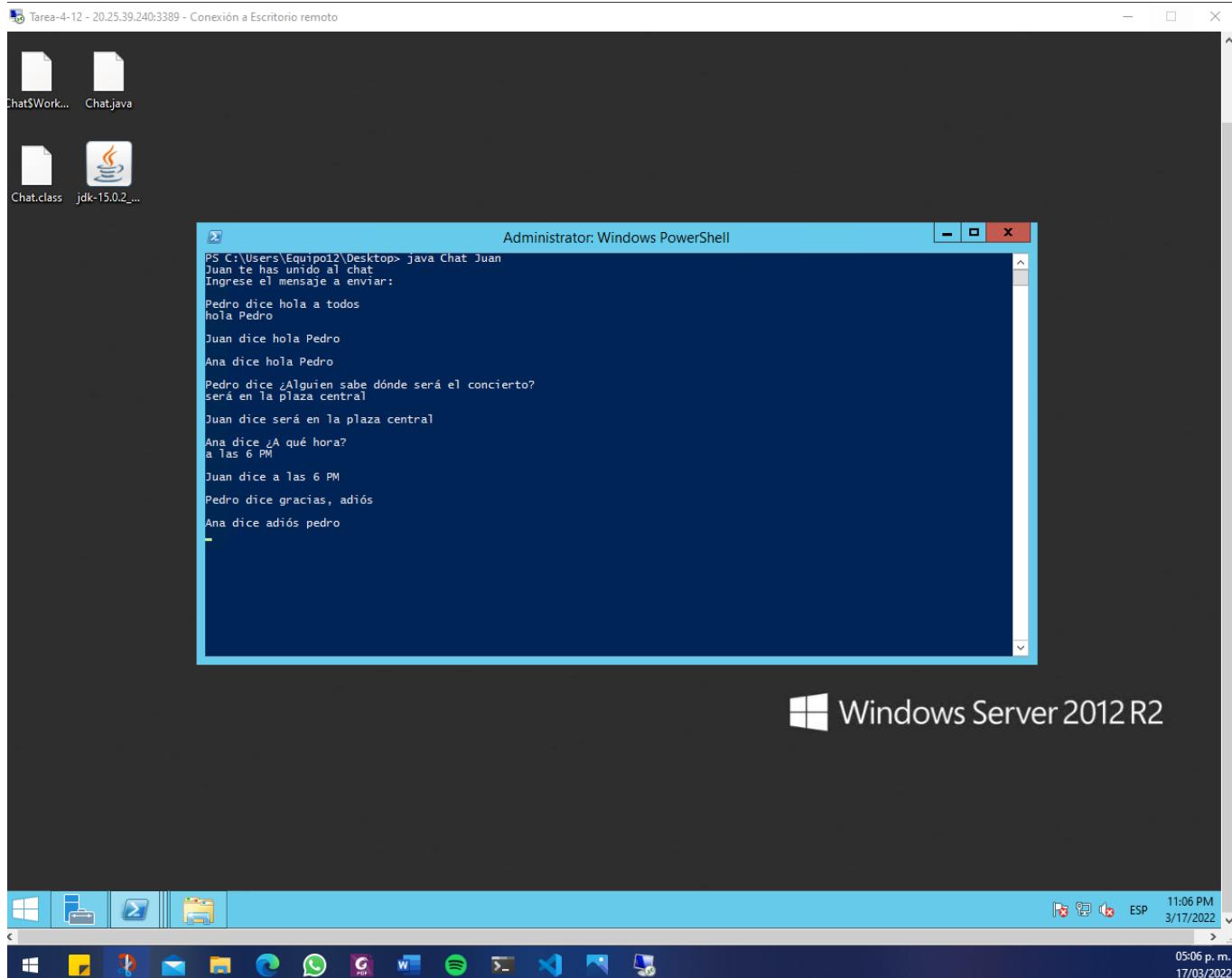


Figura 33 Chat de Juan

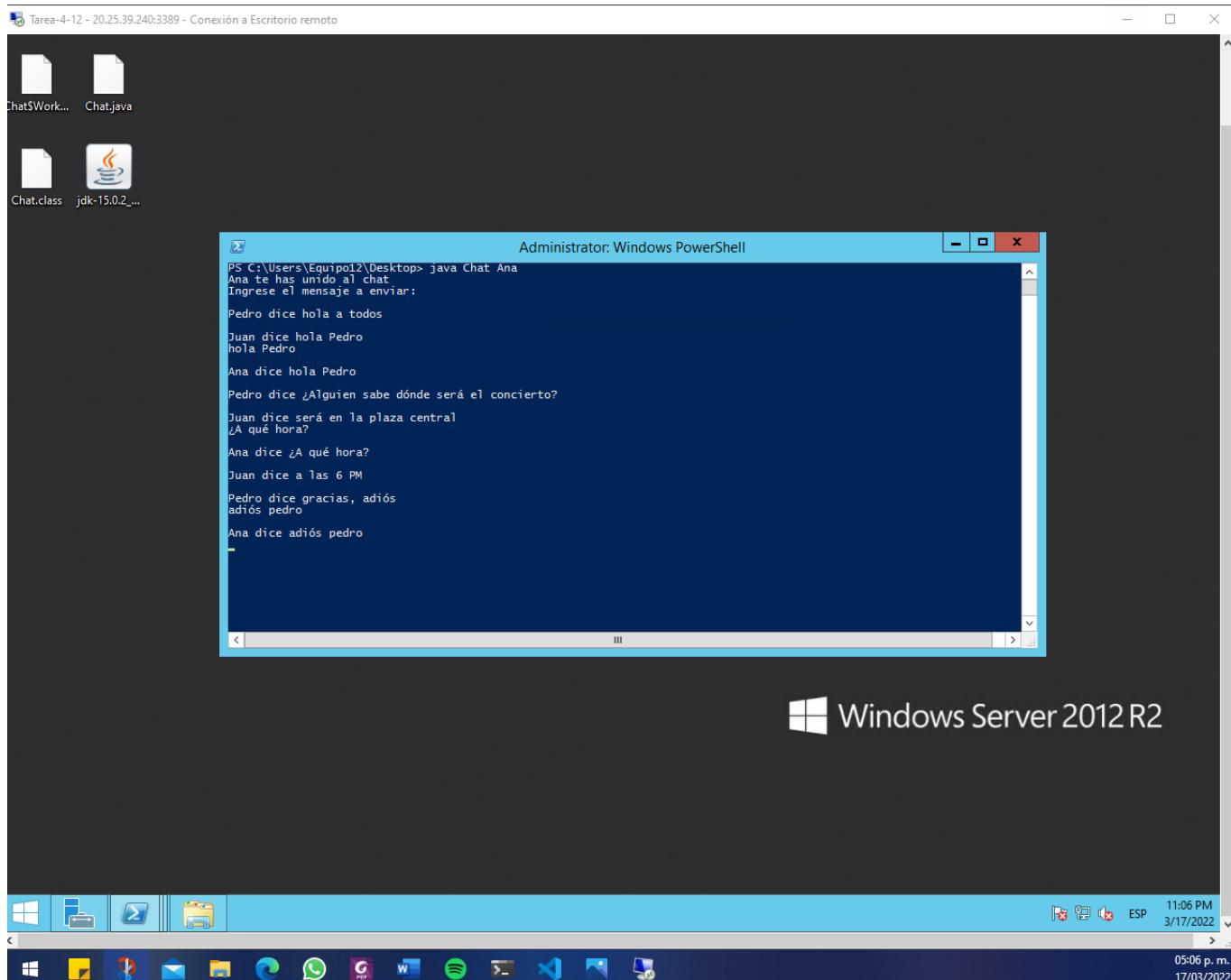


Figura 34 Chat de Ana

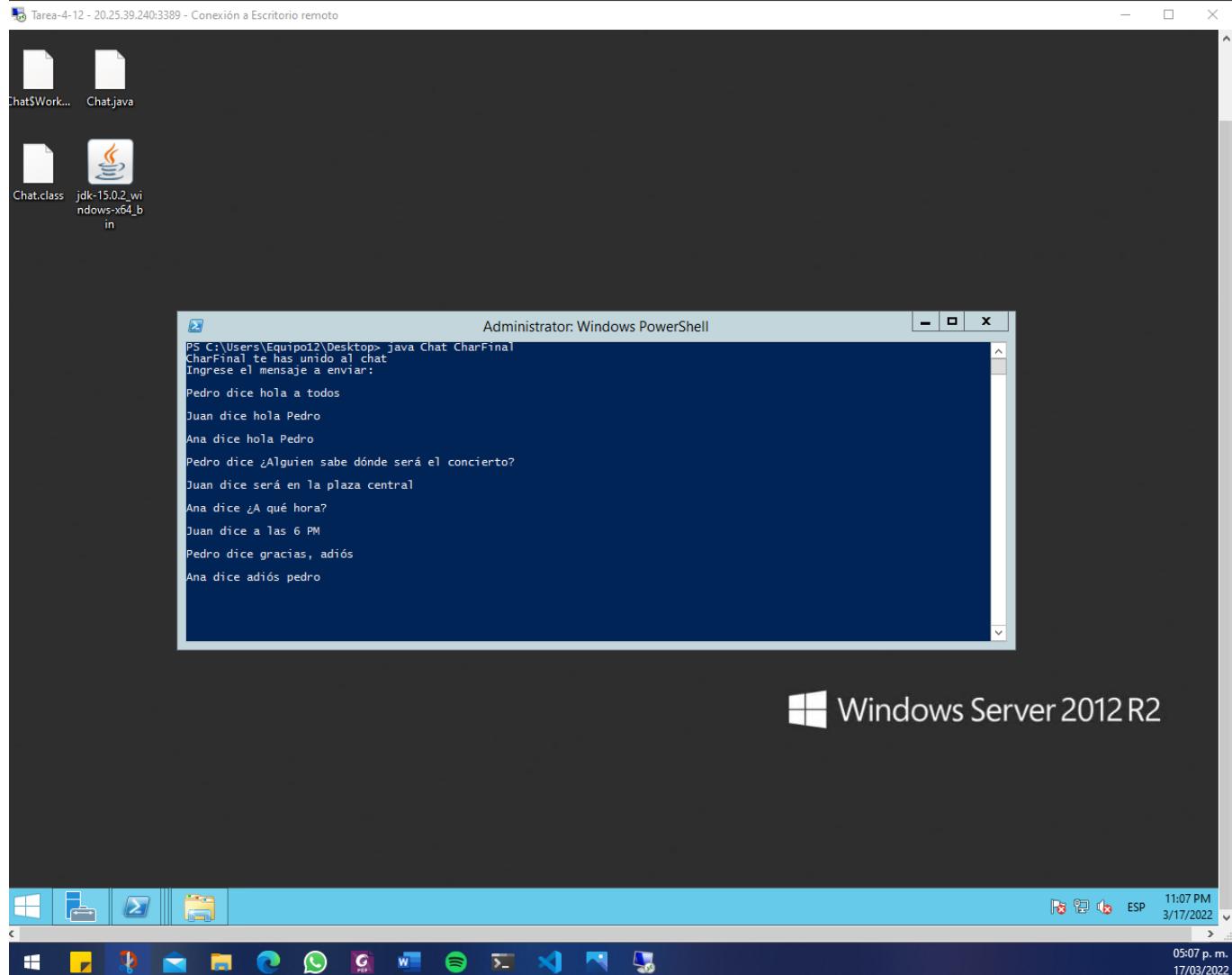


Figura 35 Conversación final entre Pedro, Juan y Ana

Concusiones

Ramírez Galindo Karina

Con esta práctica se implementaron bastantes conceptos vistos en clase, se vio un mejor control y fluidez en la arquitectura cliente-servidor utilizando sockets multicast a través del protocolo UDP. La implementación del chat permitió ver de forma precisa el flujo de los paquetes por medio de bytes. De esta manera, todos los clientes que se fueron conectando a una sala general, fueron manejados por los sockets y a su vez por hilos que fueron recibiendo y mandando flujos de información.

La ejecución fue bastante simple debido a que solo se hizo en una máquina virtual con todos los dispositivos conectados a la misma red y al mismo puerto. Lo más tardado fue la creación y configuración de la máquina virtual con Windows Server 2012 junto con la conexión remota mediante RDP, además de la instalación del JDK 15, por lo que también se tuvo que hacer una modificación en los métodos `joinGroup(SocketAddress mcastaddr, NetworkInterface netIf)` y `leaveGroup(SocketAddress mcastaddr, NetworkInterface netIf)` en el archivo Chat.java.

Por último, es importante destacar que se utilizó la codificación “Cp850” para que los caracteres especiales como acentos, signos, etc fueran mostrados en consola.

Toledo Espinosa Cristina Aline

En esta práctica nuevamente se hizo uso de las máquinas virtuales que nos permite crear Azure, estas son muy fáciles de crear y hacer funcionar, al menos en comparación con las máquinas virtuales que se pueden crear localmente con ayuda de VirtualBox, sin importar el sistema operativo que se use ya que en este caso fue Windows y lo único tardado para este caso fue la conexión remota la cual según la configuración dada a la propia MV afecta directamente al tiempo de carga y conexión. Se pudo observar también el funcionamiento de mensajes tipo multicast que como se sabe son mensajes enviados por un solo usuario (en este caso) y le llegan a TODOS los conectados a una red (en este caso al chat), que básicamente es el funcionamiento de los grupos creados en las redes sociales (como son WhatsApp o Messenger) solo que a una mucho menor escala.

Vázquez Hernández Alan Mauricio

Cuando es necesario el envío de mensajes a una gran cantidad de máquinas, utilizar conexiones TCP se vuelve lento e inefficiente, por lo que es necesario tomar otra alternativa que sea más rápida. Tal es el caso del protocolo UDP. En esta práctica fue posible explorar la comunicación entre computadoras mediante el uso de sockets multicast, los cuales envían los paquetes a todos los usuarios conectados a la red del servidor. Asimismo, también fue posible trabajar con la creación de máquinas virtuales en Azure con Windows Server, así como las configuraciones necesarias para poder ejecutar correctamente el programa (como la codificación de la consola de comandos).

Referencias

- [1] I. Inc., «Multicast: conexiones multipunto para una transmisión eficiente,» [En línea]. Available: <https://www.ionos.mx/digitalguide/servidores/know-how/multicast/>.
- [2] C. M. Kozierok, «IP Multicast Addressing,» [En línea]. Available: http://www.tcpipguide.com/free/t_IPMulticastAddressing.htm.
- [3] P. G. Carlos, «Desarrollo de sistemas Distribuidos - 4CV11 Plataforma Educativa Moodle,» [En línea]. Available: <https://m4gm.com/moodle>.