



Tarea 3-Chat Multicast

25-03-2023

Pérez Antonio David

Boleta: 2019630589

Grupo: 4CV12

Prof: Carlos Pineda Guerrero

Objetivo.....	1
Desarrollo.....	1
Conversación con Chat GPT.....	1
David Pérez Antonio.....	2
Chat GPT.....	4
Observaciones.....	7
David Pérez Antonio.....	7
Chat GPT.....	7
Nota 1:.....	8
Chat GPT.....	8
Creando máquina virtual con Windows.....	9
Paso 1 Abrimos el portal de Azure.....	9
Paso 2 se llenan los datos requeridos.....	10
Paso 3 Conectándonos con RDP.....	13
Paso 4 Montando una unidad.....	15
Ejecutando el programa.....	17
Inicio.....	17
Hugo.....	18
Paco.....	19
Luis.....	20
Conclusiones.....	21

Objetivo

Desarrollar un chat Multicast y ejecutarlo sobre Windows server en una máquina virtual en Azure.

Desarrollo

Se emplea Chat GPT para ver si puede realizar el código en un solo intento solo con la descripción de la tarea.

Conversación con Chat GPT

David Pérez Antonio

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.

El programa deberá hacer lo siguiente:

El programa creará un thread 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: "Escribe tu mensaje: " (sin las comillas), entonces se leerá del teclado el mensaje.

Se deberá enviar el mensaje a los nodos que pertenecen al grupo identificado por la IP 239.0.0.0 a través del puerto 50000. El mensaje a enviar en el datagrama deberá tener la siguiente forma:

nombre_usuario--->mensaje_ingresado

Dónde nombre_usuario es el nombre del usuario que pasó como parámetro al programa, "--->" es un separador y mensaje_ingresado es el mensaje que el usuario ingresó por el teclado.

Debido a que el protocolo UDP no garantiza el orden en que se reciben los mensajes, no se deberá enviar la longitud del mensaje y luego el mensaje; se deberá utilizar una longitud de mensaje fija. Al desplegar el mensaje recibido se deberán eliminar los espacios adicionales a la derecha (si los hay).

Para demostrar el programa se deberá utilizar los siguientes usuarios: Hugo, Paco y Luis.

Se deberá ejecutar la siguiente conversación (la cual aparece en negritas) en tres ventanas de comandos (cmd) en la máquina virtual con Windows Server 2012. En la primera ventana escribirá Hugo, en la segunda ventana escribirá Paco y en la tercera ventana escribirá Luis:

Hugo escribirá:

hola a todos

Paco escribirá:

hola Hugo

Luis escribirá:

hola Paco

hola Hugo

Paco escribirá:

¿alguien sabe dónde será el concierto?

Hugo escribirá:

será en la plaza central

Paco escribirá:

¿a qué hora?

Luis escribirá:

a las 8 PM

Paco escribirá:

gracias, adios

Luis debe escribir:

adios Paco

Los signos de interrogación y las letras acentuadas deberán desplegarse correctamente en la ventana de comandos de Windows.

Chat GPT

Entiendo que se trata de una tarea o un ejercicio para programar un chat usando sockets UDP en Java y se requiere una máquina virtual con Windows Server 2012 en Azure. A continuación, se proporciona un código de ejemplo que puede utilizarse para implementar el programa solicitado:

java

Copy code

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.InetAddress;
import java.net.MulticastSocket;
```

```
public class ChatMulticastUDP {
```

```
    private static final String MULTICAST_GROUP_IP = "239.0.0.0";
    private static final int PORT = 50000;
    private static final int MAX_PACKET_SIZE = 100; // Tamaño máximo del mensaje
```

```
    private String username;
    private InetAddress multicastGroup;
```

Desarrollo de sistemas distribuidos

```

private MulticastSocket multicastSocket;

public ChatMulticastUDP(String username) throws IOException {
    this.username = username;
    this.multicastGroup = InetAddress.getByName(MULTICAST_GROUP_IP);
    this.multicastSocket = new MulticastSocket(PORT);
    this.multicastSocket.joinGroup(this.multicastGroup);
}

private void startReceivingMessages() {
    Thread thread = new Thread(() -> {
        try {
            while (true) {
                byte[] buffer = new byte[MAX_PACKET_SIZE];
                DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
                this.multicastSocket.receive(packet);

                String receivedMessage = new String(packet.getData(), packet.getOffset(),
packet.getLength()).trim();

                String senderUsername = receivedMessage.substring(0,
receivedMessage.indexOf("--->"));

                String message = receivedMessage.substring(senderUsername.length() + 4);
                if (!senderUsername.equals(this.username)) {
                    System.out.println(senderUsername + ": " + message);
                }
            }
        } catch (IOException ex) {
            System.err.println("Error al recibir mensaje multicast: " + ex.getMessage());
        }
    });
    thread.start();
}

```

```

public void startSendingMessages() {
    try {
        DatagramPacket packet;
        String message;
        byte[] buffer;
        while (true) {
            System.out.print("Escribe tu mensaje: ");
            message = this.username + "--->" + readLine();
            buffer = message.getBytes();
            packet = new DatagramPacket(buffer, buffer.length, this.multicastGroup, PORT);
            this.multicastSocket.send(packet);
        }
    } catch (IOException ex) {
        System.err.println("Error al enviar mensaje multicast: " + ex.getMessage());
    }
}

```

```

private static String readLine() {
    StringBuilder sb = new StringBuilder();
    try {
        int ch = System.in.read();
        while (ch != -1 && ch != '\n') {
            sb.append((char) ch);
            ch = System.in.read();
        }
    } catch (IOException ex) {
        System.err.println("Error al leer de teclado: " + ex.getMessage());
    }
    return sb.toString().trim();
}

```

```

public static void main(String[] args) throws IOException {
    if (args.length != 1) {
        System.err.println("Debe proporcionar el nombre de usuario como argumento.");
        System.exit(1);
    }
    ChatMulticastUDP chat = new ChatMulticastUDP(args[0]);
    chat.startReceivingMessages();
    chat.startSendingMessages();
}
}

```

Este código define una clase ChatMulticastUDP que se encarga de recibir y enviar mensajes en un grupo multicast de direcciones IP 239.0.0.0 en el puerto 50000. Para crear una instancia de esta clase, se debe proporcionar el nombre de usuario como argumento.

El método startReceivingMessages() crea un hilo que se encarga de recibir los mensajes

Observaciones

El chat creó un programa funcional al primer intento, pero aun olvido algunas partes entre ellas los signos de interrogación y acentos no se muestran correctamente, tampoco puso la parte de que el hilo también debía desplegar el mensaje del mismo nodo dado en la instrucción: El thread desplegará el mensaje que envía el mismo nodo.

Debido a lo anterior, se harán pequeñas correcciones a mano.

David Pérez Antonio

```

private static String readLine() {
    StringBuilder sb = new StringBuilder();
    try {
        int ch = System.in.read();
        while (ch != -1 && ch != '\n') {
            sb.append((char) ch);
            ch = System.in.read();
        }
    } catch (IOException ex) {
        System.err.println("Error al leer de teclado: " + ex.getMessage());
    }
}

```



```
}  
    return sb.toString().trim();  
}
```

modifica este metodo para que se detecten correctamente las ñ, acentos y signos de interrogacion desde el teclado

Chat GPT

Para detectar correctamente los caracteres especiales como las ñ, acentos y signos de interrogación desde el teclado, se debe utilizar un `InputStreamReader` para leer el flujo de entrada del teclado y especificar la codificación adecuada.

Aquí está el método modificado:

java

Copy code

```
private static String readLine() {  
    StringBuilder sb = new StringBuilder();  
    try {  
        InputStreamReader isr = new InputStreamReader(System.in, "UTF-8");  
        int ch = isr.read();  
        while (ch != -1 && ch != '\n') {  
            sb.append((char) ch);  
            ch = isr.read();  
        }  
    } catch (IOException ex) {  
        System.err.println("Error al leer de teclado: " + ex.getMessage());  
    }  
    return sb.toString().trim();  
}
```

En este caso, se está utilizando la codificación UTF-8, que es capaz de manejar la mayoría de los caracteres especiales utilizados en diferentes idiomas, incluidos los acentos y la ñ en español.

Desarrollo de sistemas distribuidos

Nota 1:

se omite parte de la conversación que fueron demasiadas preguntas y que muchas de ellas no llevaban a ningún lado, a continuación se deja solo la parte que fue de más ayuda para resolver el problema de codificación.

Chat GPT

Para cambiar la configuración del código de página de consola (chcp) a través del registro de Windows (Regedit), sigue estos pasos:

Abre el editor de registro de Windows presionando las teclas Windows + R, escribe regedit y presiona Enter.

Navega a la siguiente clave en el árbol de navegación del registro:

sql

Copy code

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Nls\CodePage

Busca la clave llamada OEMCP en la lista de claves. Haz clic derecho sobre ella y selecciona "Modificar".

En la ventana "Editar valor DWORD", selecciona "Decimal".

Ingresa el valor correspondiente al código de página que deseas utilizar. Por ejemplo, para establecer la configuración del código de página en 65001 (UTF-8), ingresa el valor 65001. Luego, haz clic en "Aceptar" para guardar los cambios.

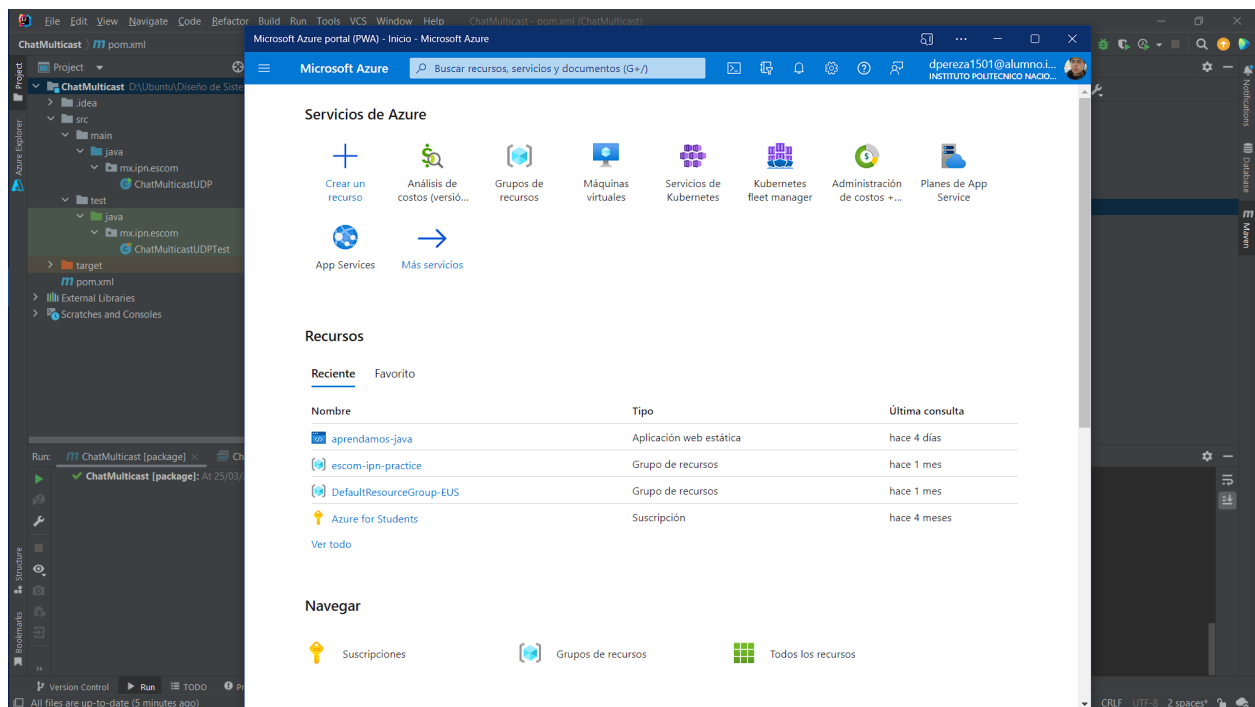
Cierra el editor de registro y reinicia la consola de PowerShell para aplicar los cambios.

Es importante tener en cuenta que cambiar la configuración del código de página de consola utilizando el editor de registro puede tener consecuencias imprevistas en otras aplicaciones que utilizan la consola de Windows. Por lo tanto, es recomendable hacer una copia de seguridad del registro antes de realizar cambios.

Creando máquina virtual con Windows

Desarrollo de sistemas distribuidos

Paso 1 Abrimos el portal de Azure



Paso 2 se llenan los datos requeridos

UDP chat using Java x Curso: Desarrollo de Sistemas Distribuidos x SIDSIS-4CV12: Tarea 3. Chat multi x Tarea 3 - Chat Multicast - Docum x +

Microsoft Azure portal (PWA) - Máquinas virtuales - Microsoft Azure

Inicio > Máquinas virtuales x

Instituto Politécnico Nacional (correo.ipn.mx)

+ Crear Cambiar al modo clásico Reservas Administrar vista Actualizar Exportar a CSV Abrir consulta

Máquina virtual de Azure

Crear una máquina virtual hospedada por Azure

Máquina virtual de Azure preestablecida

Crear una máquina virtual con valores predeterminados basados en sus cargas de trabajo

Máquina virtual de Azure Arc

Crear una nueva máquina virtual de Azure Arc en uno de los entornos que no son de Azure

Máquina virtual de Azure VMware Solution

Crear una máquina virtual de VMware hospedada por Azure

No hay máquinas virtuales para mostrar

Crear una máquina virtual que ejecute Linux o Windows. Seleccione una imagen de Marketplace o use una imagen personalizada propia.

+ Crear

Más información acerca de Windows Virtual Machines

Más información sobre Linux Virtual Machines

Enviar comentarios

UDP chat using Java x Curso: Desarrollo de Sistemas Distribuidos x SIDSIS-4CV12: Tarea 3. Chat multi x Tarea 3 - Chat Multicast - Docum x +

Microsoft Azure portal (PWA) - Crear una máquina virtual - Microsoft Azure

Inicio > Máquinas virtuales >

Crear una máquina virtual

Tamaño

Standard_B1ms - 1 vcpu, 2 GiB de memoria (USD 17.96/mes)

Ver todos los tamaños

Cuenta de administrador

Nombre de usuario

Dezksr

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

RDP (3389)

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: Discos >

Enviar comentarios

Microsoft Azure portal (PWA) - Crear una máquina virtual - Microsoft Azure

Inicio > Máquinas virtuales >

Crear una máquina virtual

default when persisting it to the cloud.

Cifrado en el host ☐

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

OS disk

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 ☒

Key management

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_B1ms).

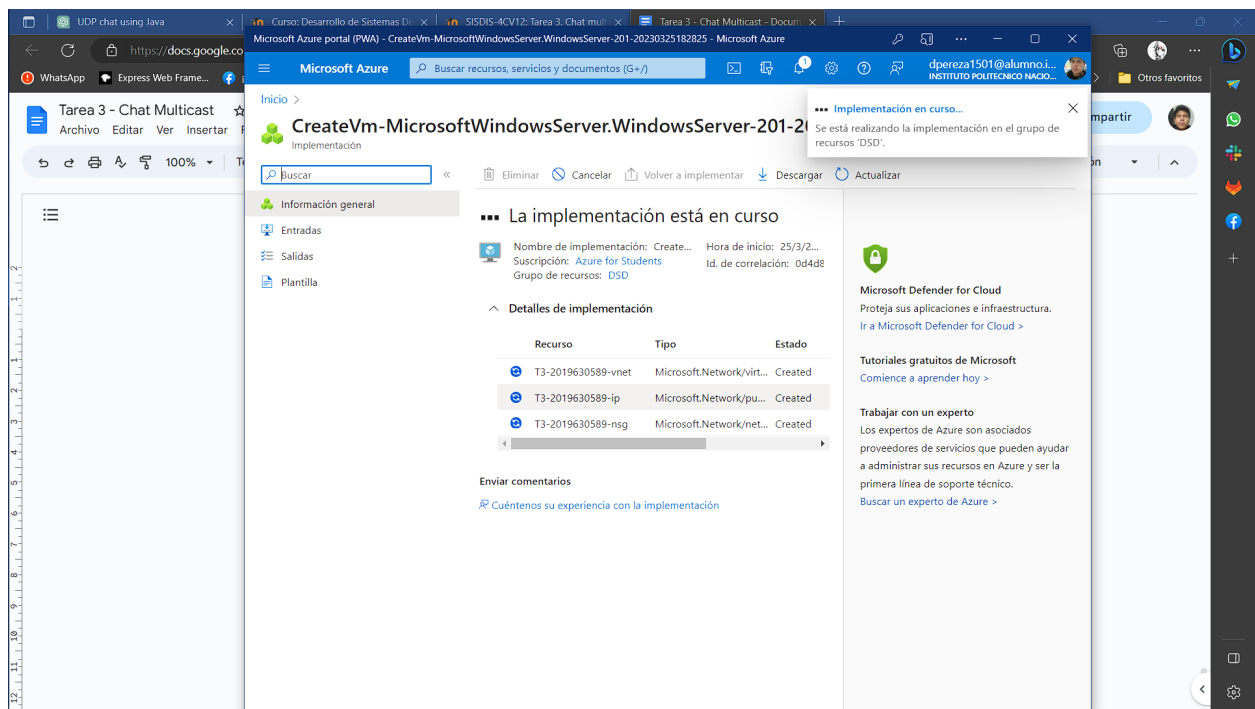
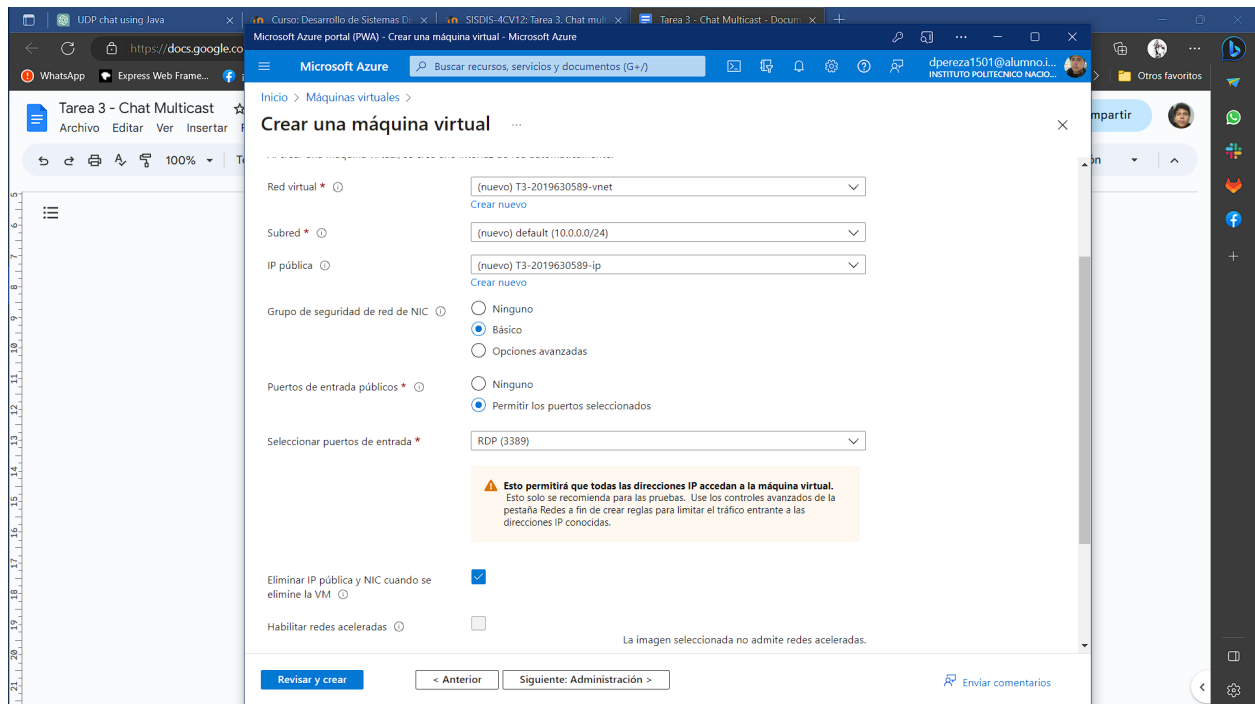
Discos de datos para T3-2019630589

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

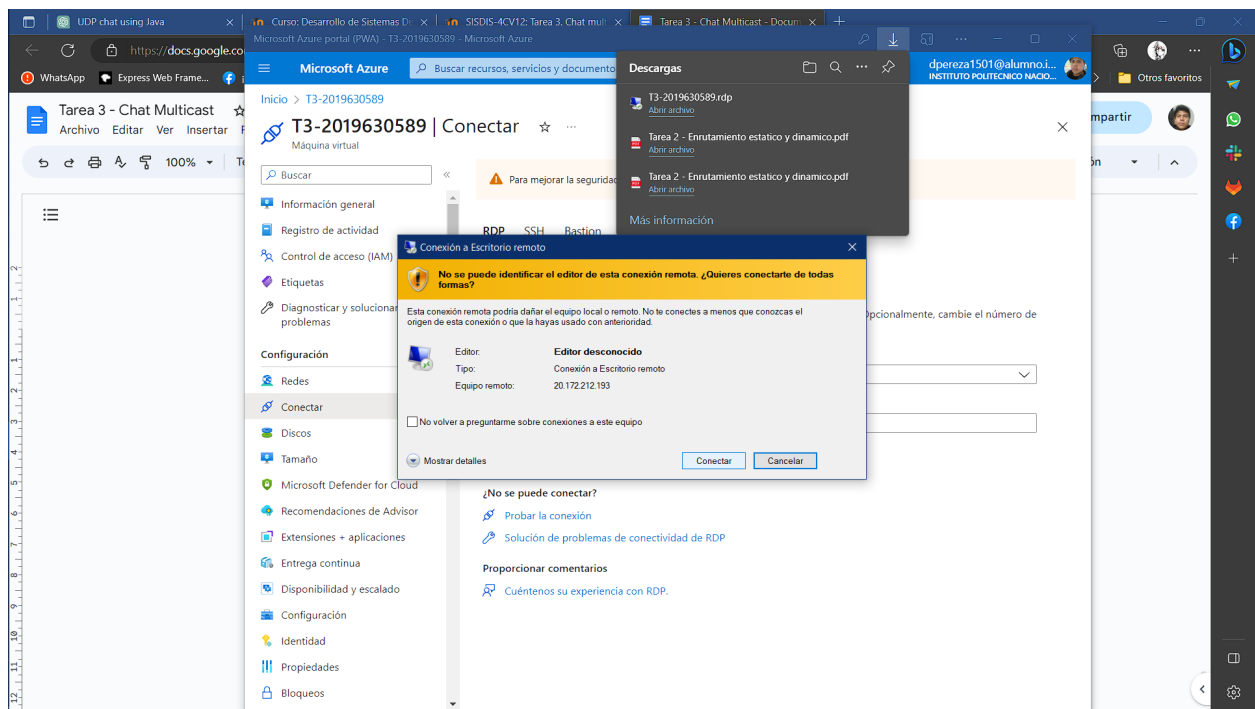
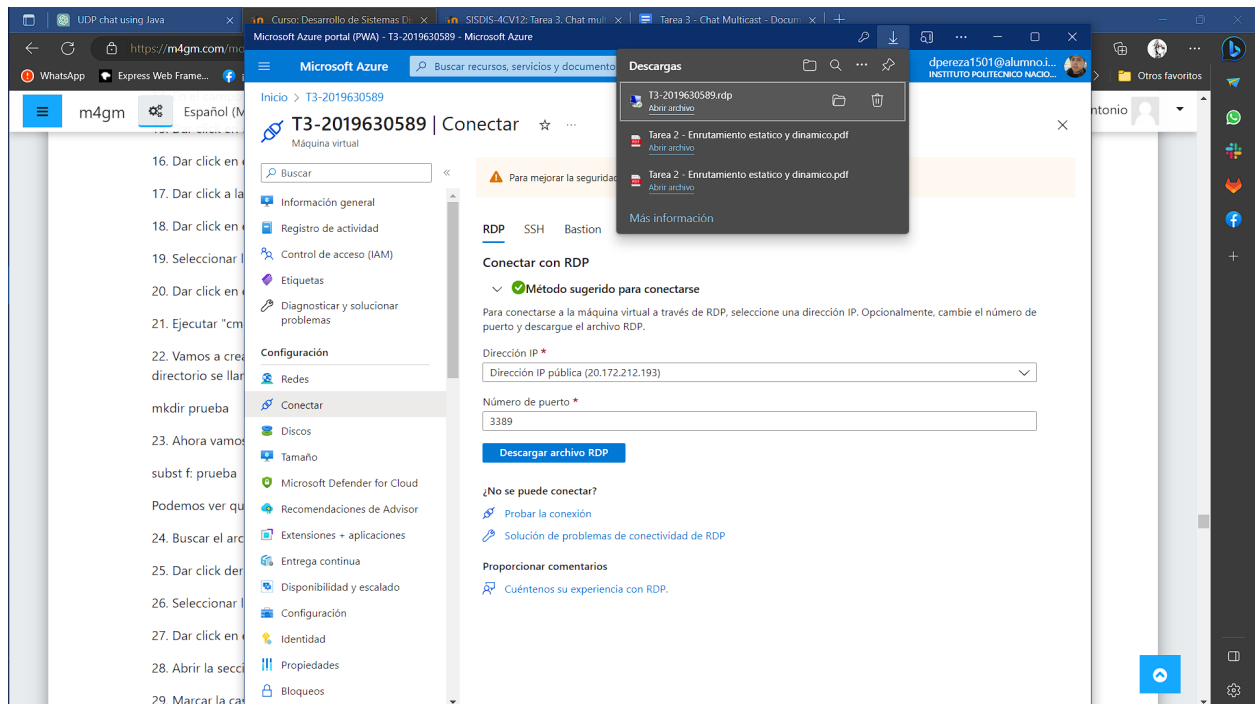
LUN	Nombre	Tamaño	Tipo de disco	Almacena...	Eliminar con VM
Crear y adjuntar un nuevo disco Asociar un disco existente					

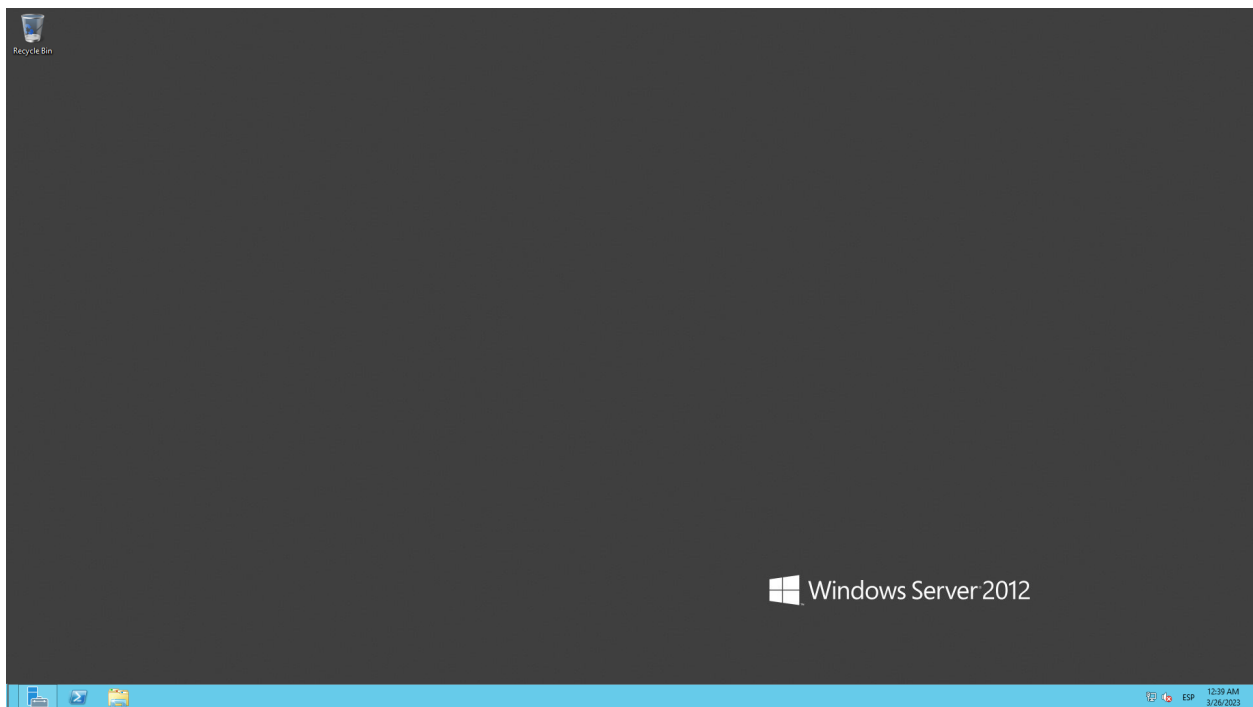
[Revisar y crear](#) [Anterior](#) [Siguiente: Redes](#)

[Enviar comentarios](#)

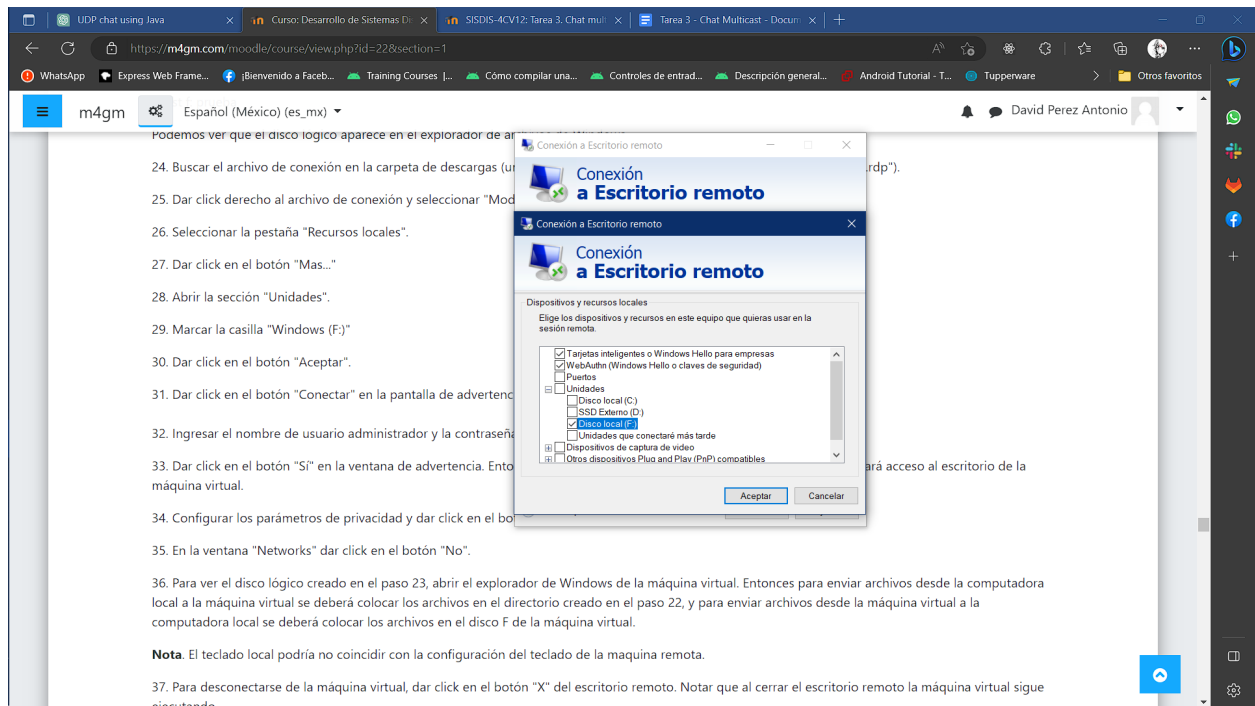


Paso 3 Conectándonos con RDP





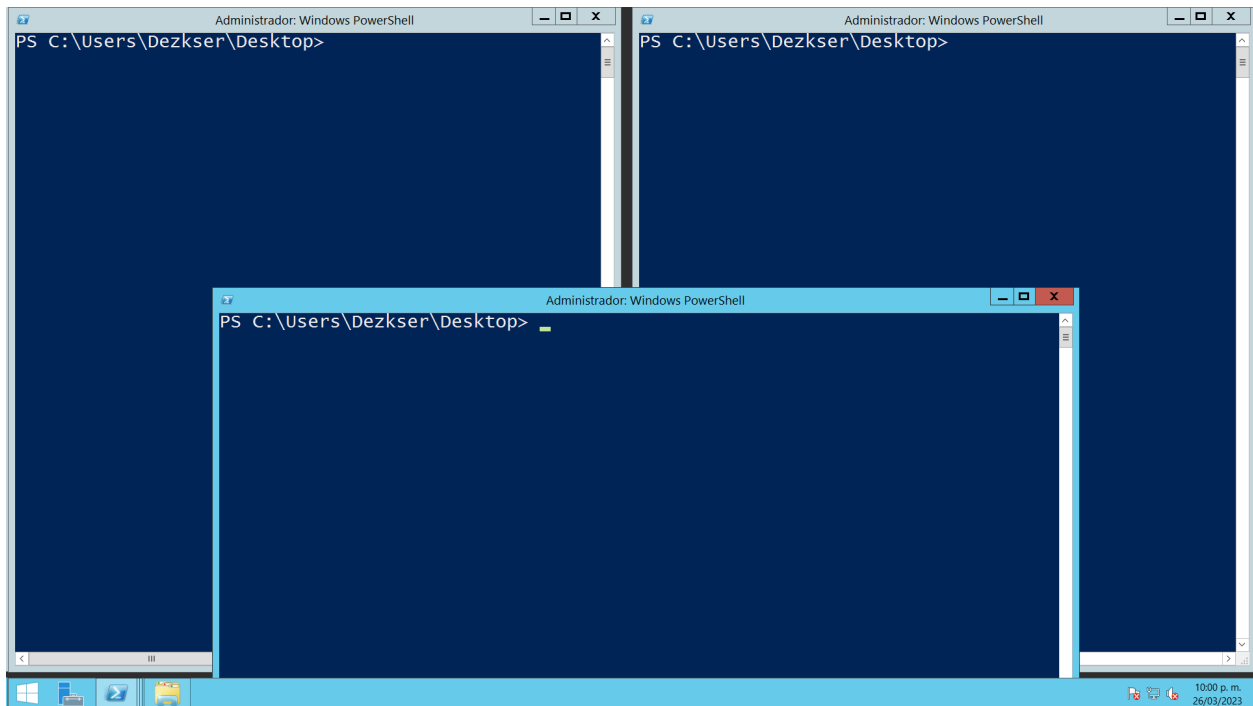
Paso 4 Montando una unidad



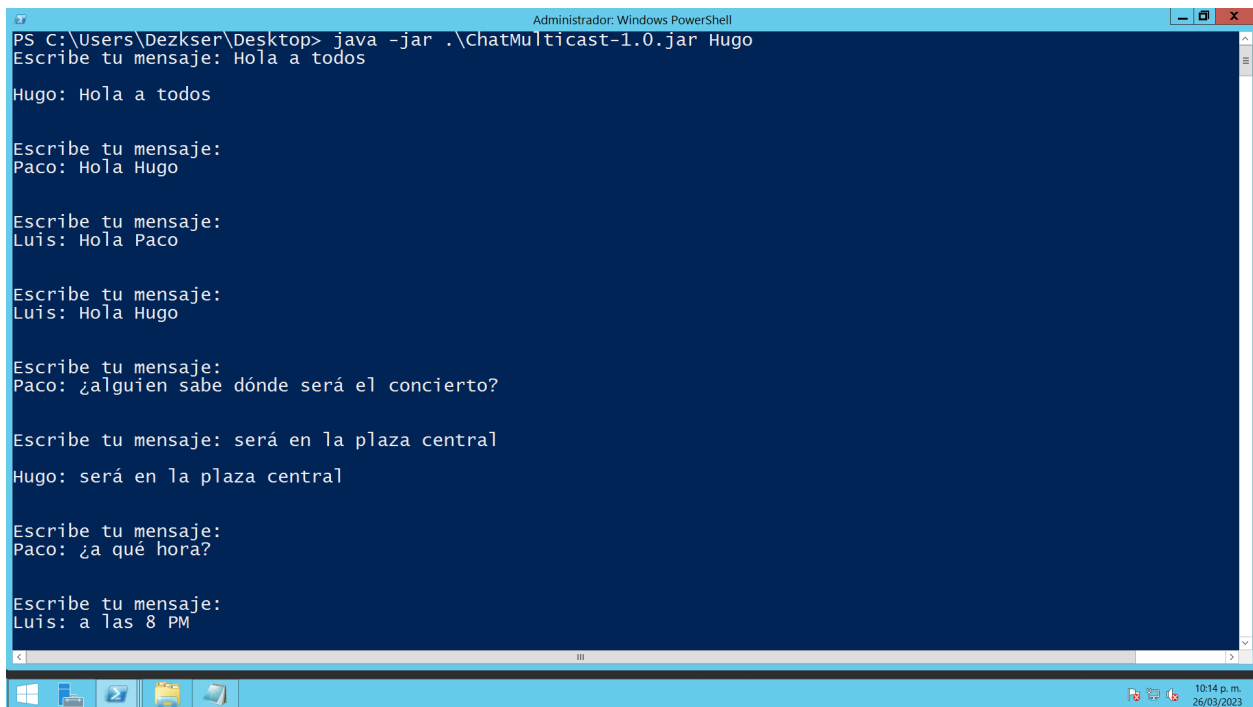
Ejecutando el programa

Inicio

Abrimos 3 terminales con un buen tamaño de letra, tras muchas preguntas a Chat GPT y búsqueda en foros se determinó que debía cambiarse un registro del sistema para que coincidiera con la codificación utilizada en el programa, tras varios intentos la mejor codificación fue CP850, esto debido a que la imagen de windows server 2012 no tenia instalado el juego de caracteres 65001 correspondientes a UTF-8, debido a esto al ejecutar el programa no se veía nada en la consola.



Hugo



Desarrollo de sistemas distribuidos

```
Administrador: Windows PowerShell

Escribe tu mensaje:
Paco: ¿alguien sabe dónde será el concierto?

Escribe tu mensaje: será en la plaza central
Hugo: será en la plaza central

Escribe tu mensaje:
Paco: ¿a qué hora?

Escribe tu mensaje:
Luis: a las 8 PM

Escribe tu mensaje:
Paco: gracias, adios

Escribe tu mensaje:
Luis: adios Paco

Escribe tu mensaje: _
```

Paco

```
Administrador: Windows PowerShell
PS C:\Users\Dezkser\Desktop> java -jar .\ChatMulticast-1.0.jar Paco
Escribe tu mensaje:
Hugo: Hola a todos

Escribe tu mensaje: Hola Hugo
Paco: Hola Hugo

Escribe tu mensaje:
Luis: Hola Paco

Escribe tu mensaje:
Luis: Hola Hugo

Escribe tu mensaje: ¿alguien sabe dónde será el concierto?
Paco: ¿alguien sabe dónde será el concierto?

Escribe tu mensaje:
Hugo: será en la plaza central

Escribe tu mensaje: ¿a qué hora?
Paco: ¿a qué hora?

Escribe tu mensaje:
Luis: a las 8 PM
```

```
Administrador: Windows PowerShell

Escribe tu mensaje:
Hugo: será en la plaza central

Escribe tu mensaje: ¿a qué hora?
Paco: ¿a qué hora?

Escribe tu mensaje:
Luis: a las 8 PM

Escribe tu mensaje: gracias, adios
Paco: gracias, adios

Escribe tu mensaje:
Luis: adios Paco

Escribe tu mensaje: _
```

Luis

```
Administrador: Windows PowerShell
PS C:\Users\Dezkser\Desktop> java -jar .\ChatMulticast-1.0.jar Luis
Escribe tu mensaje:
Hugo: Hola a todos

Escribe tu mensaje:
Paco: Hola Hugo

Escribe tu mensaje: Hola Paco
Luis: Hola Paco

Escribe tu mensaje: Hola Hugo
Luis: Hola Hugo

Escribe tu mensaje:
Paco: ¿alguien sabe dónde será el concierto?

Escribe tu mensaje:
Hugo: será en la plaza central

Escribe tu mensaje:
Paco: ¿a qué hora?

Escribe tu mensaje: a las 8 PM
Luis: a las 8 PM
```

```
Administrador: Windows PowerShell

Escribe tu mensaje:
Hugo: será en la plaza central

Escribe tu mensaje:
Paco: ¿a qué hora?

Escribe tu mensaje: a las 8 PM
Luis: a las 8 PM

Escribe tu mensaje:
Paco: gracias, adios

Escribe tu mensaje: adios Paco
Luis: adios Paco

Escribe tu mensaje: _
```

Conclusiones

La escritura del código fue lo más simple del mundo, gracias al uso del Chat en el primer intento nos dio un código funcional solo utilizando como prompt la descripción de la tarea, por otro lado el código tenía errores que no eran fáciles de detectar a simple vista, tuvieron que ser corregidos a mano para poder mostrar correctamente los caracteres en windows

server, además hubo un detalle que chat gpt no pudo explicar y que se tuvo que deducir a muchas pruebas y errores, este detalle era que windows server no tenía instalado el juego de caracteres UTF-8 por lo que cuando usaba ese tipo de codificación el programa no funcionaba correctamente, además se tuvo que cambiar la región y el idioma del servidor para tener disponible el juego de caracteres de México CP 850, también se adaptó el programa para que coincidiera con el juego de caracteres de windows server.

Se eligió CP 850 debido a que no encontré una manera de instalar el juego 65001 correspondiente a UTF-8 desde windows server, no había documentación oficial y el chat no daba nada de información útil.

Con este trabajo se concluye que el chat a pesar de ser muy bueno no puede mantener información suficiente en memoria sobre lo que está ocurriendo y tampoco puede igualar la capacidad creativa, pues todas sus respuestas se limitaban a utilizar solamente datos con los que contaba, algunos de ellos desactualizados y otros que se acercaban a lo necesario pero sin llegar a lo requerido.