**Logo

Description automatically generatedINSTITUTO POLITÉCNICO NACIONAL**

**Escuela Superior de Cómputo**

**UNIDAD DE APRENDIZAJE**:

Desarrollo de Sistemas Distribuidos

**Tarea 3. ChatMulticast, ChatGPT**

**INTEGRANTES DEL EQUIPO**

* Chávez Morones Ángel Uriel

**NOMBRE DEL PROFESOR:**

Carlos Pineda

**Parte 1.**

Creación de la máquina virtual Windows

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Contra123456!

Una captura de pantalla de una computadora

Descripción generada automáticamente con confianza media

Programa ejecutandose

Texto

Descripción generada automáticamente

Entrevista con ChatGP para la solución del problema

chat multicast, programa que cree un thread el cual recibira los mensajes del resto de los nodos. Cada mensaje recibido será desplegado en la pantalla. El thread desplegará el mensaje que envia el mismo nodo dentro del main, que esta en un ciclo infinito se desplegará el siguientee prompt: "Escribe tu mensaje" Los signos de interrogacion y las letras deberan desplegarse correctamente en la ventana de comandos de Windows. El nimbre de usuario se pasara como parámetro,en Java

ChatGPT:¡Hola! Para crear un programa que permita la comunicación multicast y visualice los mensajes recibidos en pantalla, podemos utilizar la clase **MulticastSocket** de Java, así como la clase **Thread** para manejar el hilo de recepción de mensajes.

Aquí te dejo un ejemplo de cómo podría ser el código:

javaCopy code

import java.io.IOException; import java.net.DatagramPacket; import java.net.InetAddress; import java.net.MulticastSocket; import java.nio.charset.StandardCharsets; import java.util.Scanner; public class ChatMulticast { private static final String MULTICAST\_IP = "230.0.0.1"; // dirección IP multicast utilizada private static final int PORT = 4444; // puerto utilizado para la comunicación multicast private static final int BUFFER\_SIZE = 1024; // tamaño del buffer para los mensajes public static void main(String[] args) throws IOException { if (args.length != 1) { System.err.println("Debe indicar su nombre de usuario como parámetro."); System.exit(1); } String username = args[0]; // nombre de usuario // creamos el socket multicast MulticastSocket socket = new MulticastSocket(PORT); InetAddress group = InetAddress.getByName(MULTICAST\_IP); socket.joinGroup(group); // creamos el hilo de recepción de mensajes Thread receiverThread = new Thread(() -> { while (true) { byte[] buffer = new byte[BUFFER\_SIZE]; DatagramPacket packet = new DatagramPacket(buffer, buffer.length); try { socket.receive(packet); String message = new String(packet.getData(), StandardCharsets.UTF\_8).trim(); // mostramos el mensaje recibido en pantalla System.out.println(message); } catch (IOException e) { e.printStackTrace(); } } }); receiverThread.start(); // leemos los mensajes que escriba el usuario desde la consola Scanner scanner = new Scanner(System.in); while (true) { System.out.print("Escribe tu mensaje: "); String message = scanner.nextLine(); // agregamos el nombre de usuario al mensaje message = username + ": " + message; // creamos el paquete de datos a enviar byte[] buffer = message.getBytes(StandardCharsets.UTF\_8); DatagramPacket packet = new DatagramPacket(buffer, buffer.length, group, PORT); socket.send(packet); } } }

Este programa crea un socket multicast y se une a un grupo multicast específico, donde enviará y recibirá mensajes. Además, crea un hilo que se encargará de recibir los mensajes y mostrarlos en pantalla.

Dentro del **main**, se lee el nombre de usuario como argumento y se crea el hilo de recepción de mensajes. Luego, se entra en un ciclo infinito donde se lee desde la consola el mensaje que quiere enviar el usuario, se le agrega el nombre de usuario y se envía al grupo multicast.

Para ejecutar este programa, debes compilarlo y ejecutarlo con el nombre de usuario como argumento:

phpCopy code

javac ChatMulticast.java java ChatMulticast <nombre de usuario>

Espero que esto te sea de ayuda. Si tienes alguna duda o necesitas más información, no dudes en preguntar.

No me desplego bien los signos ni los acentos

ChatGPT: Si los signos de interrogación y las letras con acentos no se están desplegando correctamente en la ventana de comandos de Windows, puede deberse a que la consola está utilizando un conjunto de caracteres diferente al que estás utilizando en tu código.

Para solucionarlo, puedes intentar especificar explícitamente el conjunto de caracteres utilizado en la consola utilizando la clase **Console**, que proporciona un método **charset()** que devuelve el conjunto de caracteres utilizado por la consola.

Aquí te dejo un ejemplo de cómo podrías modificar el código para usar el conjunto de caracteres adecuado:

javaCopy code

import java.io.IOException; import java.net.DatagramPacket; import java.net.InetAddress; import java.net.MulticastSocket; import java.nio.charset.Charset; import java.util.Scanner; public class ChatMulticast { private static final String MULTICAST\_IP = "230.0.0.1"; // dirección IP multicast utilizada private static final int PORT = 4444; // puerto utilizado para la comunicación multicast private static final int BUFFER\_SIZE = 1024; // tamaño del buffer para los mensajes public static void main(String[] args) throws IOException { if (args.length != 1) { System.err.println("Debe indicar su nombre de usuario como parámetro."); System.exit(1); } String username = args[0]; // nombre de usuario // creamos el socket multicast MulticastSocket socket = new MulticastSocket(PORT); InetAddress group = InetAddress.getByName(MULTICAST\_IP); socket.joinGroup(group); // creamos el hilo de recepción de mensajes Thread receiverThread = new Thread(() -> { while (true) { byte[] buffer = new byte[BUFFER\_SIZE]; DatagramPacket packet = new DatagramPacket(buffer, buffer.length); try { socket.receive(packet); String message = new String(packet.getData(), Charset.forName(System.console().charset().name())).trim(); // mostramos el mensaje recibido en pantalla System.out.println(message); } catch (IOException e) { e.printStackTrace(); } } }); receiverThread.start(); // leemos los mensajes que escriba el usuario desde la consola Scanner scanner = new Scanner(System.in); while (true) { System.out.print("Escribe tu mensaje: "); String message = scanner.nextLine(); // agregamos el nombre de usuario al mensaje message = username + ": " + message; // creamos el paquete de datos a enviar byte[] buffer = message.getBytes(Charset.forName(System.console().charset().name())); DatagramPacket packet = new DatagramPacket(buffer, buffer.length, group, PORT); socket.send(packet); } } }

En este ejemplo, se utiliza el método **Charset.forName(System.console().charset().name())** para obtener el conjunto de caracteres utilizado por la consola y utilizarlo al convertir las cadenas de bytes a cadenas de caracteres y viceversa.

Espero que esto solucione tu problema. Si tienes alguna otra pregunta, no dudes en preguntar.

y como se compilaria?

ChatGPT: Para compilar este programa, sigue los siguientes pasos:

1. Abre un editor de texto y copia y pega el código en un archivo de nombre **ChatMulticast.java**.
2. Abre una ventana de línea de comandos y navega hasta la carpeta donde se encuentra el archivo **ChatMulticast.java**.
3. Asegúrate de tener instalado el kit de desarrollo de Java (JDK) en tu sistema. Si no lo tienes, descárgalo e instálalo desde la página oficial de Oracle: <https://www.oracle.com/java/technologies/javase-downloads.html>
4. Ejecuta el siguiente comando para compilar el archivo **ChatMulticast.java**:

Copy code

javac ChatMulticast.java

1. Si todo ha ido bien, se habrá generado un archivo **ChatMulticast.class**.
2. Ahora puedes ejecutar el programa utilizando el siguiente comando:

phpCopy code

java ChatMulticast <nombre\_de\_usuario>

Reemplaza **<nombre\_de\_usuario>** con el nombre que desees utilizar como usuario en el chat. Por ejemplo:

Copy code

java ChatMulticast johndoe

¡Y eso es todo! Ahora puedes utilizar el chat multicast.

cual seria la configuracion de la consola para que pueda ver los acentos y signos de interrogacion?

ChatGPT: La configuración de la consola dependerá del sistema operativo que estés utilizando. En general, para poder ver correctamente los caracteres con acentos y los signos de interrogación en la consola, deberás asegurarte de que se está utilizando el conjunto de caracteres adecuado para tu idioma.

A continuación, te doy algunas indicaciones generales que pueden ayudarte a configurar correctamente la consola en distintos sistemas operativos:

* **Windows**: Abre la consola de comandos y haz clic derecho en la barra superior de la ventana. Selecciona "Propiedades" y luego ve a la pestaña "Página de códigos". Asegúrate de que se esté utilizando el conjunto de caracteres adecuado para tu idioma, como "Windows-1252" para español o "UTF-8" para varios idiomas.
* **Linux y macOS**: Abre la terminal y ve al menú "Preferencias". Selecciona la pestaña "Codificación" y asegúrate de que se esté utilizando el conjunto de caracteres adecuado para tu idioma, como "UTF-8" para varios idiomas.

Ten en cuenta que la configuración de la consola puede variar según la distribución de Linux o la versión de macOS que estés utilizando. Si tienes dificultades para configurar la consola correctamente, te recomiendo que busques información específica para tu sistema operativo y versión.

La consola ya esta configurada pero el siguiente codigo no me imprime con los signos de interrogación y acentuación import java.io.IOException; import java.net.DatagramPacket; import java.net.InetAddress; import java.net.MulticastSocket; import java.nio.charset.Charset; import java.util.Scanner; public class ChatMulticast { private static final String MULTICAST\_IP = "230.0.0.1"; // dirección IP multicast utilizada private static final int PORT = 4444; // puerto utilizado para la comunicación multicast private static final int BUFFER\_SIZE = 1024; // tamaño del buffer para los mensajes public static void main(String[] args) throws IOException { if (args.length != 1) { System.err.println("Debe indicar su nombre de usuario como parámetro."); System.exit(1); } String username = args[0]; // nombre de usuario // creamos el socket multicast MulticastSocket socket = new MulticastSocket(PORT); InetAddress group = InetAddress.getByName(MULTICAST\_IP); socket.joinGroup(group); // creamos el hilo de recepción de mensajes Thread receiverThread = new Thread(() -> { while (true) { byte[] buffer = new byte[BUFFER\_SIZE]; DatagramPacket packet = new DatagramPacket(buffer, buffer.length); try { socket.receive(packet); String message = new String(packet.getData(), Charset.forName(System.console().charset().name())).trim(); // mostramos el mensaje recibido en pantalla System.out.println(message); } catch (IOException e) { e.printStackTrace(); } } }); receiverThread.start(); // leemos los mensajes que escriba el usuario desde la consola Scanner scanner = new Scanner(System.in); while (true) { System.out.print("Escribe tu mensaje: "); String message = scanner.nextLine(); // agregamos el nombre de usuario al mensaje message = username + ": " + message; // creamos el paquete de datos a enviar byte[] buffer = message.getBytes(Charset.forName(System.console().charset().name())); DatagramPacket packet = new DatagramPacket(buffer, buffer.length, group, PORT); socket.send(packet); } } } Podrías modificarlo para que me imprima cprrectamente lo que necesito?

ChatGPT: Sí, para imprimir correctamente los caracteres con acentos y signos de interrogación en la consola, debes asegurarte de que el encoding que estás utilizando sea el adecuado para el idioma que estás utilizando.

En el siguiente código, he modificado la línea donde se lee el mensaje recibido en el hilo de recepción, para que utilice el encoding "UTF-8", que es el más comúnmente utilizado para idiomas que utilizan caracteres acentuados y signos de interrogación. Además, he modificado la línea donde se obtiene el encoding de la consola, para que sea más precisa.

Prueba a reemplazar el código que tienes actualmente por el siguiente:

javaCopy code

import java.io.IOException; import java.net.DatagramPacket; import java.net.InetAddress; import java.net.MulticastSocket; import java.nio.charset.StandardCharsets; import java.util.Scanner; public class ChatMulticast { private static final String MULTICAST\_IP = "230.0.0.1"; // dirección IP multicast utilizada private static final int PORT = 4444; // puerto utilizado para la comunicación multicast private static final int BUFFER\_SIZE = 1024; // tamaño del buffer para los mensajes public static void main(String[] args) throws IOException { if (args.length != 1) { System.err.println("Debe indicar su nombre de usuario como parámetro."); System.exit(1); } String username = args[0]; // nombre de usuario // creamos el socket multicast MulticastSocket socket = new MulticastSocket(PORT); InetAddress group = InetAddress.getByName(MULTICAST\_IP); socket.joinGroup(group); // creamos el hilo de recepción de mensajes Thread receiverThread = new Thread(() -> { while (true) { byte[] buffer = new byte[BUFFER\_SIZE]; DatagramPacket packet = new DatagramPacket(buffer, buffer.length); try { socket.receive(packet); String message = new String(packet.getData(), StandardCharsets.UTF\_8).trim(); // mostramos el mensaje recibido en pantalla System.out.println(message); } catch (IOException e) { e.printStackTrace(); } } }); receiverThread.start(); // leemos los mensajes que escriba el usuario desde la consola Scanner scanner = new Scanner(System.in, System.console().charset().name()); while (true) { System.out.print("Escribe tu mensaje: "); String message = scanner.nextLine(); // agregamos el nombre de usuario al mensaje message = username + ": " + message; // creamos el paquete de datos a enviar byte[] buffer = message.getBytes(StandardCharsets.UTF\_8); DatagramPacket packet = new DatagramPacket(buffer, buffer.length, group, PORT); socket.send(packet); } } }

Espero que te sea de ayuda y que ahora puedas ver correctamente los caracteres en la consola.

Me muestra un espacio vacio en lugar del caracter

ChatGPT: Entiendo. Esto se debe a que probablemente la fuente utilizada en la consola no tiene un carácter definido para el símbolo en cuestión. Una solución es utilizar una fuente que incluya el conjunto completo de caracteres necesarios.

Otra opción es utilizar la codificación UTF-8 para la transferencia de datos en lugar de la codificación predeterminada, que puede no admitir ciertos caracteres. Para ello, debes modificar la creación del **DatagramPacket** para que use la codificación UTF-8, de la siguiente manera:

arduinoCopy code

byte[] buffer = message.getBytes(StandardCharsets.UTF\_8); DatagramPacket packet = new DatagramPacket(buffer, buffer.length, group, PORT); socket.send(packet);

Además, para imprimir correctamente los caracteres recibidos, puedes forzar la codificación UTF-8 al crear la cadena de texto, de la siguiente manera:

scssCopy code

String message = new String(packet.getData(), StandardCharsets.UTF\_8).trim(); System.out.println(message);

Con estas modificaciones, deberías poder enviar y recibir correctamente caracteres como los acentos y signos de interrogación.

Ya esta implementado y aun asi no me deja imprimir los caracteres

ChatGPT: Si después de implementar las modificaciones que te sugerí aún tienes problemas para imprimir correctamente los caracteres, es posible que el problema esté relacionado con la configuración regional de tu sistema. Te sugiero verificar que tu sistema esté configurado para utilizar la codificación de caracteres adecuada para tu idioma y región.

También puedes intentar especificar la codificación de caracteres al iniciar la máquina virtual de Java agregando el argumento **-Dfile.encoding=UTF-8** al comando de ejecución de tu programa.

Por ejemplo, si estás ejecutando tu programa desde la línea de comandos, puedes usar el siguiente comando:

phpCopy code

java -Dfile.encoding=UTF-8 ChatMulticast <nombre\_de\_usuario>

De esta manera, estás asegurando que la máquina virtual de Java esté configurada para utilizar la codificación UTF-8 en todo momento.

sigue sin funcionar

ChatGPT: Si después de intentar las soluciones sugeridas aún no puedes imprimir correctamente los caracteres acentuados y los signos de interrogación, podrías intentar lo siguiente:

1. Verificar que la fuente utilizada en la consola tenga soporte para los caracteres que deseas imprimir. Si no es así, puedes intentar cambiar la fuente a una que los admita.
2. Especificar la codificación de caracteres directamente en la línea de comandos, utilizando la opción **-Dfile.encoding**, como te mencioné anteriormente. En lugar de UTF-8, también podrías intentar con otras codificaciones, como ISO-8859-1 o Windows-1252, que son comunes en algunos sistemas.
3. Verificar que los mensajes enviados y recibidos se estén manipulando correctamente en cada paso del proceso. Para hacerlo, podrías agregar algunos mensajes de depuración en tu código para imprimir los valores de las cadenas en cada etapa del proceso.
4. Si todo lo demás falla, podrías intentar ejecutar tu programa en otro sistema para ver si el problema persiste. Si no lo hace, podría ser un problema específico de configuración en tu sistema.

se tiene que poner alguna configuracion especial para leer signos de interrogacion y acentuacion directamente desde consola con java?

ChatGPT: Sí, en algunos sistemas es posible que tengas que configurar la codificación de caracteres de la consola para poder imprimir correctamente los signos de interrogación y los caracteres acentuados desde Java.

La forma de hacerlo depende del sistema operativo que estés utilizando. Por ejemplo, en Windows puedes abrir la consola de comandos (cmd) y seleccionar "Propiedades" en el menú de la ventana. En la pestaña "Página de código", puedes seleccionar la codificación de caracteres adecuada para tu idioma (por ejemplo, "Windows-1252" para español). En otros sistemas operativos, como Linux o macOS, puedes utilizar variables de entorno para especificar la codificación de caracteres, como **LANG** o **LC\_ALL**.

En Java, puedes configurar la codificación de caracteres que se utiliza en la consola mediante la propiedad del sistema **file.encoding**. Por ejemplo, si estás utilizando Windows-1252, puedes ejecutar tu programa Java con la siguiente opción:

Copy code

java -Dfile.encoding=windows-1252 TuClasePrincipal

Con esta opción, Java utilizará la codificación de caracteres adecuada para imprimir correctamente los signos de interrogación y los caracteres acentuados desde la consola.

sigue sin funcionar

ChatGPT: Si sigues teniendo problemas para imprimir correctamente los caracteres en la consola, puede haber otras causas que estén afectando la salida de caracteres.

Una posible causa es que la fuente utilizada por la consola no soporte los caracteres que estás intentando imprimir. En este caso, podrías probar a cambiar la fuente de la consola a una que soporte los caracteres necesarios.

Otra posible causa es que el sistema operativo esté utilizando una codificación de caracteres diferente a la que estás utilizando en Java. En este caso, podrías tratar de forzar la codificación de caracteres en Java a la misma que utiliza el sistema operativo.

También es posible que el problema esté en el entorno de desarrollo que estás utilizando. Por ejemplo, si estás utilizando Eclipse, puede ser necesario configurar la codificación de caracteres en las preferencias de la aplicación.

Si ninguna de estas soluciones funciona, puede que necesites buscar más información específica para tu sistema operativo y entorno de desarrollo para solucionar el problema.

Conclusiones

Gracias a esta práctica podemos concluir lo siguiente:

1. La implementación de un chat multithread es una forma efectiva de permitir que varios usuarios se comuniquen entre sí en tiempo real.
2. La utilización de una máquina Azure proporciona una plataforma escalable y confiable para alojar una aplicación de chat multithread.
3. Es importante considerar los aspectos sobre permisos directamente con la máquina en Azure.
4. La experiencia práctica en la implementación de aplicaciones multithread en una plataforma en la nube como Azure puede ser una habilidad valiosa para el desarrollo de aplicaciones en línea.
5. La configuración de la consola para la aceptación de caracteres especiales.

En general, la práctica de un chat multithread en una máquina Azure ofrece una experiencia valiosa para comprender y abordar desafíos en el desarrollo de aplicaciones en línea, permitiendo que los desarrolladores adquieran habilidades y conocimientos para enfrentar desafíos en el desarrollo de aplicaciones escalables y seguras en la nube.