T3A1: ENTENDIENDO LOS SOCKETS

Ejercicio 1: Ejecución del código

● **Ejecuta el código del cliente y después el del servidor.¿Qué ocurre?¿Por qué?**

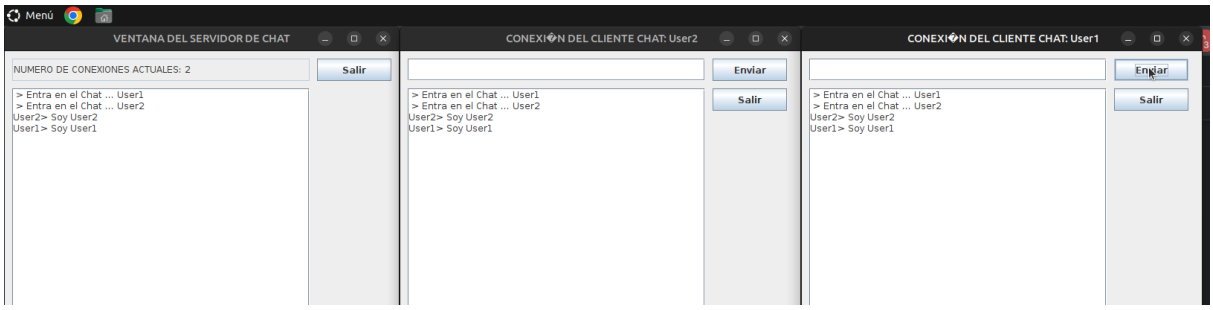
R: Da error, ya que el cliente no puede encontrar al servidor porque no está activo.

● **¿Qué protocolo de comunicación se está utilizando UDP o TCP?**

R: Se está utilizando el protocolo TCP.

● **Ejecuta el servidor y varias instancias de cliente (lee la siguiente página para saber cómo hacerlo), utiliza el chat y haz captura.**

R:



Ejercicio 2: Comprender el Código del Cliente

● **¿Cuál es el propósito de la clase ClienteChat en este código?**

R: Generar una ventana que permitirá al usuario conectarse a una instancia de la clase ServidorChat donde podrá enviar mensajes bajo un nombre de usuario.

● **Explica el flujo de comunicación entre el cliente y el servidor.¿Cómo se envían y reciben los mensajes?**

R: Cuando se instancia una clase de ClienteChat, se intenta establecer conexión con el socket: s = new Socket("localhost", puerto). Después, tanto DataInputStream como DataOutputStream utilizarán este socket como punto de conexión con el servidor para recibir y enviar datos mediante las instancias correspondientes "fentrada", "fsalida".

Cuando un cliente pulsa el botón "Enviar", se recoge el contenido del JTextField "mensaje" añadiéndolo al String "texto" tras el nombre del usuario, se borra el contenido de "mensaje" y se pasa al DataOutputStream "fsalida" mediante el método writeUTF(texto). El servidor recibe esta información a través de su DataInputStream "fentrada" y lo almacena en su JTextField correspondiente y finalmente envía el contenido de este a los clientes recorriendo con un bucle for las CONEXIONES de ServidorChat.

● **¿Qué hace el método ejecutar() en la clase ClienteChat?**

R: Guardar en la variable "texto" el contenido de del DataInputStream "fentrada" para luego plasmarlo en la "textarea1" hasta que se cierra ese flujo de entrada, es decir, hasta que se cierra el servidor.

Dicho de otra manera, comprueba continuamente si hay un nuevo mensaje.

● **¿Qué sucede si el cliente intenta enviar un mensaje después de que se ha desconectado del servidor?**

R: No puede, ya que al cerrar el servidor aparece un JDialog que avisa de esto y se cierra el programa.

Ejercicio 3: Comprender el Código del Servidor

● **¿Cuál es el papel de la clase HiloServidor en el servidor?**

R: Manejar la cantidad de clientes que se conectan

● **Explica el propósito de la matriz tabla en la clase ServidorChat. ¿Cómo se utiliza para controlar las conexiones?**

R: Es una matriz de Sockets que almacena las instancias de sockets de cad cliente.

● **Modifica el código del servidor para permitir un número máximo diferente de conexiones simultáneas.¿Qué cambios debes hacer en el código?**

R: Asignarle un mayor valor a la variable "MAXIMO" de la clase ServidorChat y a la matriz de sockets "tabla[]"

● **¿Qué sucede si un cliente se desconecta abruptamente sin salir**

adecuadamente?

R: Que no sale el mensaje de que el usuario ha abandonado el chat y no se actualiza la cantidad de conexiones actuales, manteniendo al que ha salido.

Ejercicio 4: Escenarios de Error y Manejo de Excepciones

● **¿Qué sucede si el servidor se cierra inesperadamente mientras hay clientes conectados?**

R: Aparece un JDialog que informa de que se ha perdido la conexión con el servidor

● **¿Qué excepciones maneja el código del cliente y del servidor? ¿Cómo se manejan estas excepciones?**

R: IOException: se manejan mostrando un JDialog con un mensaje de error.

SocketException: con un break evitando que se ejecute todo el bucle.