DANIEL CASTELAO

Tarea PSP – Desarrollo de un chat con arquitectura cliente/servidor

2º DESARROLLO DE APLICACIONES MULTIPLATAFORMA

CPR Daniel Castelao

Curso 2023 – 2024

Ángel Castiñeira Durán

El objetivo es desarrollar un sistema de chat cliente/servidor mediante sockets en TCP/IP.

El servidor

- Permitirá conectar varios clientes de chat simultáneamente, hasta un máximo de 10.
- Dispondrá de una única sala de chat a la que se conectarán todos los clientes.
- Cada usuario tendrá un nickname que se le solicitará antes de establecer la conexión con el servidor.
- Mostrará por pantalla todos los mensajes que se reciban desde los clientes a medida que van llegando, indicando "nickname: mensaje..."
- Cada mensaje que se reciba será reenviado a todos los clientes, incluyendo el nickname correspondiente.
- Al arrancar el servidor, se solicitará el puerto por lo que se establecerá la conexión.
- Cada vez que se conecte un nuevo cliente, se indicará por pantalla: Nuevo cliente conectado (nickname) Actualmente hay x usuarios conectados.
- Mientras no se conecte ningún usuario, o si todos los clientes se desconectan, se mostrará el mensaje "Ningún cliente conectado".
- Si el servidor se cierra, todos los clientes cerrarán adecuadamente sus conexiones tras mostrar el mensaje "El servidor se desconectó".

Los clientes

- Al arrancar se solicitará la dirección IP y el puerto de conexión del servidor. A continuación se solicitará el nickname que se empleará para identificar los mensajes del usuario, y se realizará la conexión con el servidor.
- Una vez conectado, se mostrarán los mensajes recibidos en la pantalla.
- Si el usuario escribe un mensaje, será enviado al servidor para su reenvío a todos los demás clientes.
- El cliente admitirá un comando /bye que hará que se cierre la conexión con el servidor y salga del programa.

- Cada vez que un nuevo cliente se conecta: en el cliente aparecerá un mensaje indicando "Conectado a la sala de chat" se mostrará a todos los participantes a mensaje "nickname acaba de conectarse la este chat."
- Cada vez que un nuevo cliente se desconecta, se mostrará a todos los participantes a mensaje "nickname dejó este chat."
- Cada vez que se escribe un mensaje, a todos los clientes se les mostrará el mensaje con el formato "nickname: mensaje"
- En cualquier caso, se deberán controlar los posibles errores y mostrar los correspondientes mensajes de manera controlada.

Protocolo

1. Se inicia el servidor y escucha peticiones entrantes.
2. Un cliente introduce su nombre de usuario y una dirección IP y puerto al que quiere conectarse.
3. El cliente envía una petición de conexión al servidor.
4. El Servidor recibe una petición de un nuevo cliente y comprueba si no se supera el número máximo de clientes conectados. Como no es el caso, el cliente se conecta al servidor y el servidor escribe al nuevo cliente un mensaje de bienvenida.
5. El cliente lee el mensaje de bienvenida y lo muestra por pantalla.
6. El cliente escribe un primer mensaje indicando el nombre de usuario.
7. El Servidor lee el primer mensaje del nuevo cliente y escribe a todos los clientes, menos a este, un mensaje indicando el nombre de usuario del nuevo cliente conectado. Por último, el servidor muestra por consola el número de clientes conectados (si hay más de uno).
8. Los clientes leen los mensajes y los muestran por pantalla.
9. Un cliente escribe un mensaje, se le concatena automáticamente a ese mensaje su nombre de usuario y muestra por pantalla el mensaje completo.
10. El servidor lee el mensaje completo del cliente y lo muestra por consola.
11. El servidor escribe al resto de clientes el mensaje recibido.
12. Los clientes leen el mensaje y lo muestran por pantalla.
13. Un nuevo cliente introduce su nombre de usuario y la dirección IP y puerto al que desea conectarse.
14. El cliente envía una petición de conexión al servidor.
15. El servidor recibe una petición de un nuevo cliente y comprueba si no se supera el número máximo de clientes conectados. Como sí que se supera, el cliente se conecta, el servidor escribe un mensaje a ese cliente indicando que no se puede conectar y lo desconecta. El nuevo cliente lee el mensaje y lo muestra por pantalla.
16. Uno de los clientes conectados escribe el comando de desconexión y se le añade a este mensaje el nombre de usuario automáticamente.
17. El servidor lee el comando de desconexión con el nombre de usuario del cliente y este desconecta a ese cliente.
18. El Servidor escribe a los clientes restantes un mensaje indicando la desconexión de un cliente y el nombre de usuario de este. Por último, el servidor muestra por consola el número de clientes conectados (si hay más de uno).
19. Los clientes leen el mensaje y lo muestran por pantalla.
20. El servidor recibe por consola el comando de cerrar el servidor.
21. El servidor escribe un mensaje a todos los clientes indicando la desconexión del Servidor.
22. El servidor desconecta a todos los clientes.
23. Los clientes leen el mensaje y lo muestran por pantalla.

Repositorio

Puedes encontrar el repositorio de GitHub con el código correspondiente aquí. Realmente los únicos archivos relevantes son **`Cliente1_v2.java`** y **`Servidor_v3.java`**. También, destacar que, las clases utilizadas para la interfaz gráfica son la de `IU.java` y

`IU_inicioSesion.java`. Los demás archivos son prototipos que fui haciendo para lograr el resultado final. Estos no los borré con el fin de reflejar el proceso seguido.