Desarrollo de un chat con arquitectura cliente/servidor

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

El servidor

- Permitirá conectar varios clientes de chat simultaneamente, 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 el 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 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** el mensaje *"nickname acaba de conectarse la este chat."*
- Cada vez que un nuevo **cliente** se **desconecta**, se mostrará a **todos** los participantes el **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. Iniciamos el servidor y le decimos en qué puerto alojarse.
2. El **servidor** espera por conexiones.
3. Iniciamos el **cliente** y le decimos a qué IP y puerto conectarse, así como cuál será nuestro *nickname*.
4. El **servidor** comprueba que el número de usuarios conectados no sea mayor que el número máximo.
    a. Si **no** lo supera lo añade a la lista de clientes y espera un mensaje del cliente para asignarlo como *nickname*.
5. El **servidor** hace un "broadcast" anunciando la conexión del nuevo cliente y el número de clientes conectados en ese momento.
6. El **servidor** se queda a la espera de mensajes.
7. Si el mensaje no es nulo y no es "/bye" el **servidor** hace un "broadcast" del mensaje a todos los clientes.
    a. De ser el mensaje "/bye" cierra la conexión con ese cliente. Mandando al resto de clientes un mensaje diciendo que un usuario se ha desconectado.
8. Si otro **cliente** quiere conectarse pero excede el número máximo de conexiones de usuario el **servidor** le enviará un mensaje al **cliente** diciendo que la sala está llena.
9. Si el **servidor** se detiene todos los clientes se desconectan y cierran su ejecución.