

EJERCICIO2_12. CREACIÓN DE UN CHAT CON TCP

Un servidor de chat que atiende a múltiples clientes.

El servidor del chat puede atender a varios clientes a la vez, cada cliente será atendido en un hilo: en este hilo se recibirán los mensajes de un cliente **y se enviarán al resto de miembros del chat.**

El programa servidor (hilo primario) define el número máximo de conexiones que admite e irá controlando los clientes que actualmente estén conectados. Para ello utiliza **un objeto de la clase ComunHilos que será compartido por todos los hilos. Fijarse en esta diferencia con supuestos anteriores, en este objeto tenemos la referencia de todos los sockets!!**

Este objeto de tipo ComunHilos contiene los siguientes atributos:

- **int CONEXIONES**, almacena el número de conexiones de clientes. Se incrementa en uno con cada conexión aceptada y nos sirve de índice para ir llenando un array de sockets de los clientes conectados.
- **int ACTUALES**, Almacena el número de clientes conectados en este momento. Cada vez que se desconecta **un cliente se resta uno a este atributo.**
- **int MAXIMO**, Atributo con el número máximo de conexiones que se atienden al mismo tiempo.
- **Socket tabla []=new Socket[MAXIMO]**, Array que almacena los sockets de cada uno de los clientes, así se podrá enviar la conversación del chat a todos cada vez que uno envía un mensaje.
- **String mensajes**: Contiene los mensajes del chat.

El programa servidor (hilo primario servidor)

Define un variable con el número máximo de conexiones que atiende.

Crea un ServerSocket para atender peticiones de conexión al chat.

Crea un array de sockets para llevar el control de los clientes conectados.

Crea un objeto de tipo ComunHilos donde actualiza todos sus atributos.

Dentro de un ciclo donde se controlan el número de conexiones, cada vez que un cliente solicita una conexión al chat se crea un nuevo socket, el nuevo socket creado se almacena en el array de sockets, se incrementa el número de conexiones y las conexiones actuales. Y se lanza un hilo nuevo para gestionar los mensajes de ese cliente.

Cada vez que se conecta un nuevo cliente al chat, se lanza un hilo de **la clase HiloServidorChat** para atender a ese cliente, el hilo recibe dos objetos uno el socket, distinto para cada cliente y el objeto de tipo **ComunHilos compartido por todos los hilos**, en donde tenemos la referencia a el array de sockets instanciados por el hilo primario.

La tarea del hilo HiloServidorChat

Deberá crear el flujo de entrada para poder leer desde el socket recibido los mensajes que escriba su cliente.

La tarea, contenida en el método run() del hilo, consistirá en:

Primero enviar todos los mensajes que hay actualmente en el chat a su cliente, para ello tenemos **en el objeto de tipo ComunHilos un método EnviarMensajesTodos()**

A continuación en un ciclo se escuchan los mensajes que su cliente envía (es decir, escribe en el chat, para el cliente el chat es el objeto socket que tiene conectado a su HiloServidorChat), el cliente envía mensajes repetitivamente y para finalizar y desconectarse envía un '*'.

El texto que el cliente envía se añade al atributo mensajes del objeto compartido por todos los HiloServidorChat, el método EnviarMensajesTodos() se encarga de ello.