

Programación de Servicios y Procesos. Sockets

Partiendo de lo que hemos visto de sockets TCP, se pide:

- Desarrollar un sistema de chat en el que se puedan conectar múltiples clientes.
 - Un cliente necesitará identificarse con un nombre cuando se conecte al chat. Es decir, cuando arranque el cliente, su primer argumento de ejecución será el nombre del cliente (*nombre_cliente*). Este argumento es obligatorio y sin él no debería conectarse al servidor.
 - Cuando un cliente se conecte al chat, el servidor le enviará un mensaje de bienvenida:
"Bienvenido al chat"
 - A este mensaje de bienvenida, el cliente responderá automáticamente con un
"Gracias, soy *nombre_cliente*"
 - Cada cliente podrá, por su cuenta, enviar cuantos mensajes quiera.
 - Cuando un cliente *c1* envíe un mensaje al chat, ese mensaje será broadcasteado a todos los otros clientes conectados al chat sin necesidad de recibirlo él mismo.
 - El servidor imprimirá en su pantalla de ejecución todos los mensajes enviados por todos los clientes identificando qué cliente ha enviado cada mensaje.
 - Bola extra: permitir enviar mensajes "privados" entre clientes.
-
- Fecha de entrega límite: **2 de febrero** (gmq.psp2019@gmail.com)
 - Recuerda enviar solo el código fuente
 - El encoding del código fuente debe ser UTF-8. ¡Debe compilar en todas las arquitecturas!
 - La clase del servidor será *ServerChatApplication* dentro del paquete *com.gmq.psp.sockets.chat.server*
 - La clase del cliente será *ClientChatApplication* dentro del paquete *com.gmq.psp.sockets.chat.client*
 - Antes de enviar la práctica asegúrate de que cumple todos los puntos indicados