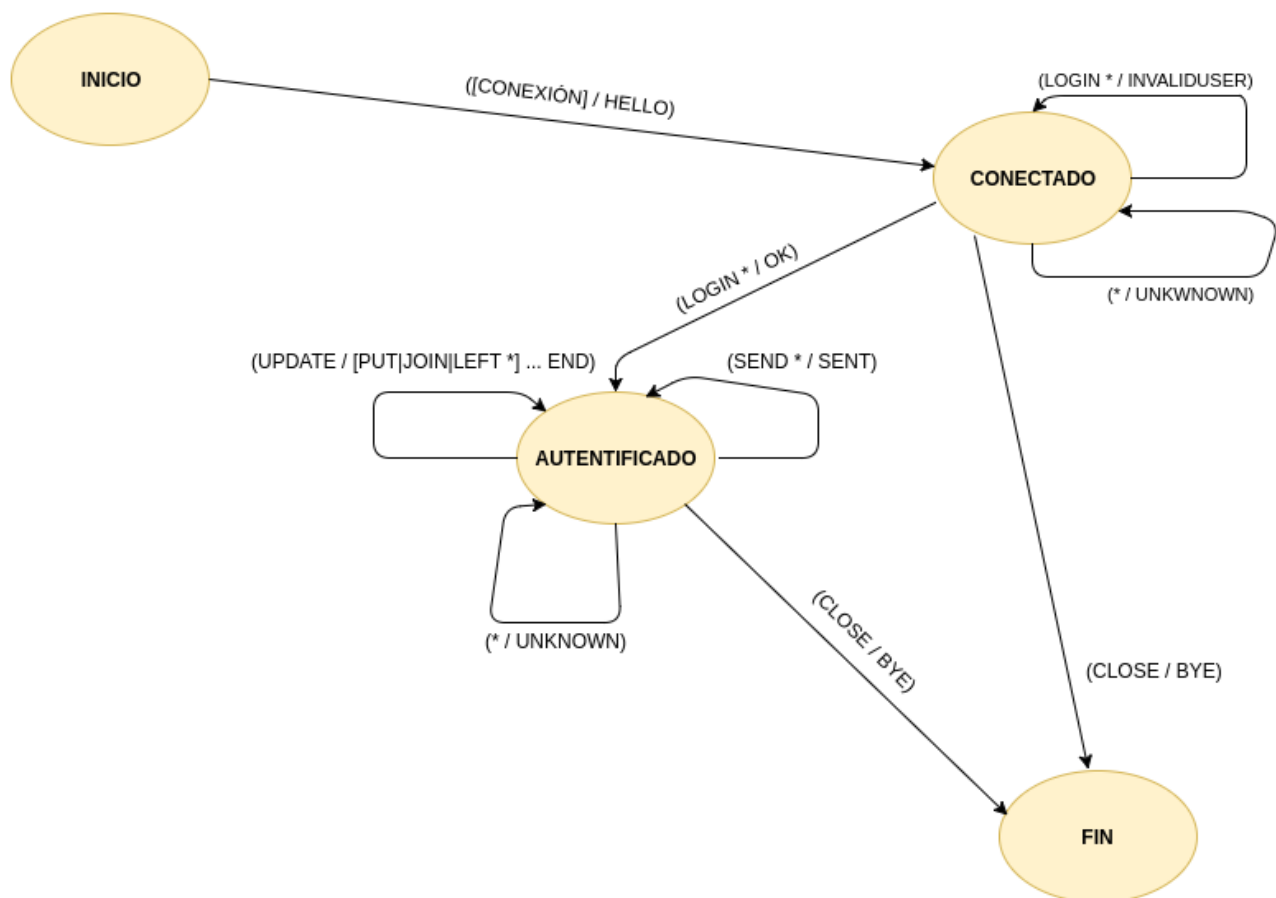


Práctica 2

Descripción de la aplicación

Nosotros hemos realizado una aplicación de chat en la que todos los clientes hablan en una sala pública a través del servidor. Dicho servidor se encarga de controlar que no se repita ningún nombre de usuario y de enviar toda la información pertinente a cada uno de los clientes desde el momento de su conexión hasta que abandonan el chat. Los clientes pueden hablar en el chat, en el que cada mensaje recibido irá precedido por el nombre de usuario y la hora de recepción (mm:ss) de dicho mensaje. Además, dispondrán de una lista para ver los usuarios que se encuentran conectados.

Diagrama de estados del servidor



Mensajes que intervienen
Cliente

Código	Cuerpo	Descripción
100	LOGIN *	Solicita entrar a la sala con el nombre de usuario especificado en el cuerpo.
200	UPDATE	Solicita una actualización de los mensajes/eventos del chat que no ha recibido todavía.
300	SEND *	Envía el mensaje especificado en el cuerpo.
400	CLOSE	El cliente informa de que va a cerrar la conexión con el servidor.

Servidor

Código	Cuerpo	Descripción
50	HELLO	Mensaje de bienvenida al servidor.
101	OK	El nombre de usuario es correcto.
102	INVALIDUSER	El nombre de usuario no es correcto
201	[PUT * JOIN * LEFT *]	El servidor informa de un mensaje nuevo (PUT), un usuario ha entrado (JOIN) o un usuario ha salido (LEFT)
202	END	Fin de una actualización (UPDATE).
301	SENT	El mensaje proporcionado ha sido enviado.
401	BYE	El servidor ha cerrado la conexión correctamente con el cliente
000	UNKNOWN	El servidor no ha sido capaz de interpretar el mensaje

Capturas de algunas funcionalidades

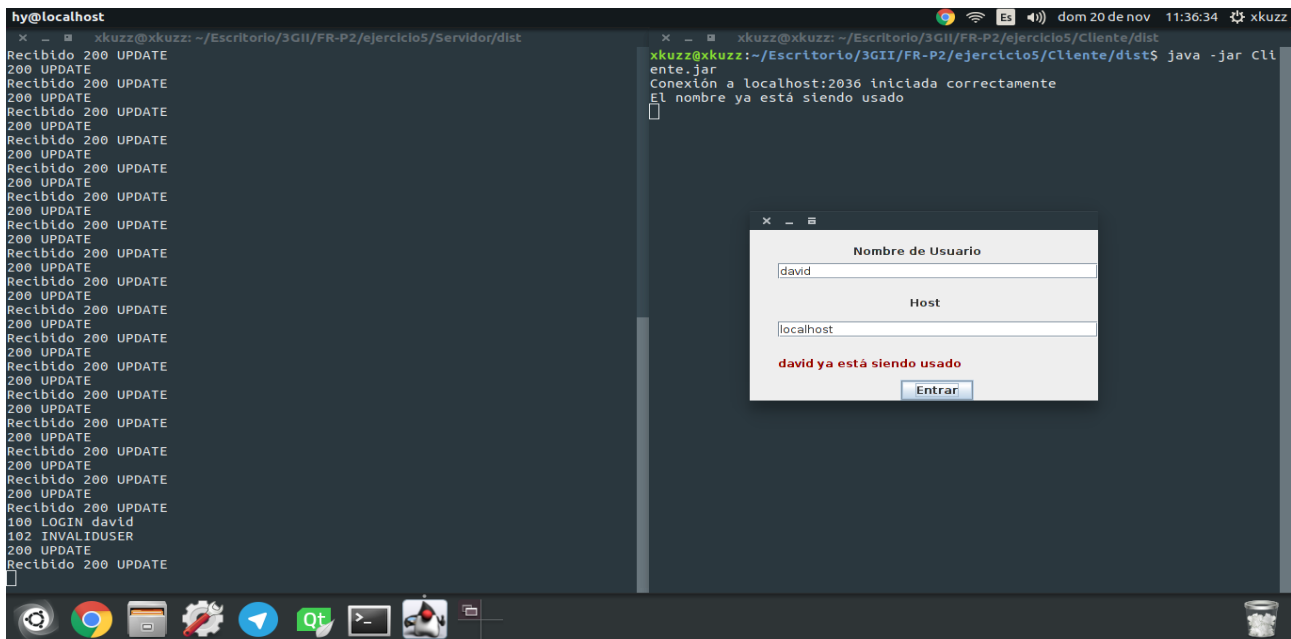


Figura 1: Ponemos un nombre de Usuario que ya está siendo usado. Tras enviarse el nombre de usuario el servidor comprueba la lista de usuarios conectados y envía el mensaje 102 INVALIDUSER

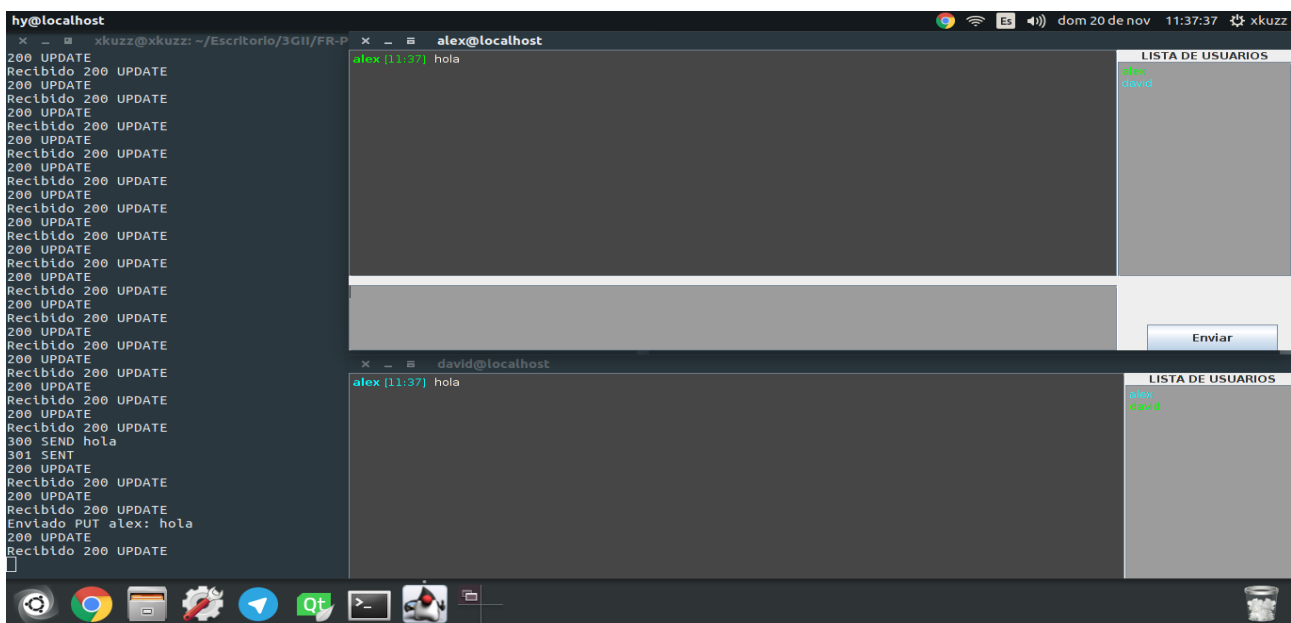


Figura 2: El usuario alex escribe el mensaje hola y pulsa Enviar, el socket envía 300 SEND hola al servidor, y este mira la lista de todos los usuarios que no son el remitente del mensaje y lo añade en sus buffers de salida. Cuando se reciba un 200 UPDATE del cliente el servidor le enviará el mensaje. La interfaz gráfica se encarga de mostrar el mensaje en la pantalla del remitente nada más enviarlo.