

# Practica 2 - Ejercicio 5

## Protocolo de aplicación - Tchat

María del Mar Ruiz Martín

Francisco Luque Sánchez

Doble Grado de Ingeniería Informática y Matemáticas

December 4, 2015

### 1 Descripción de la aplicación

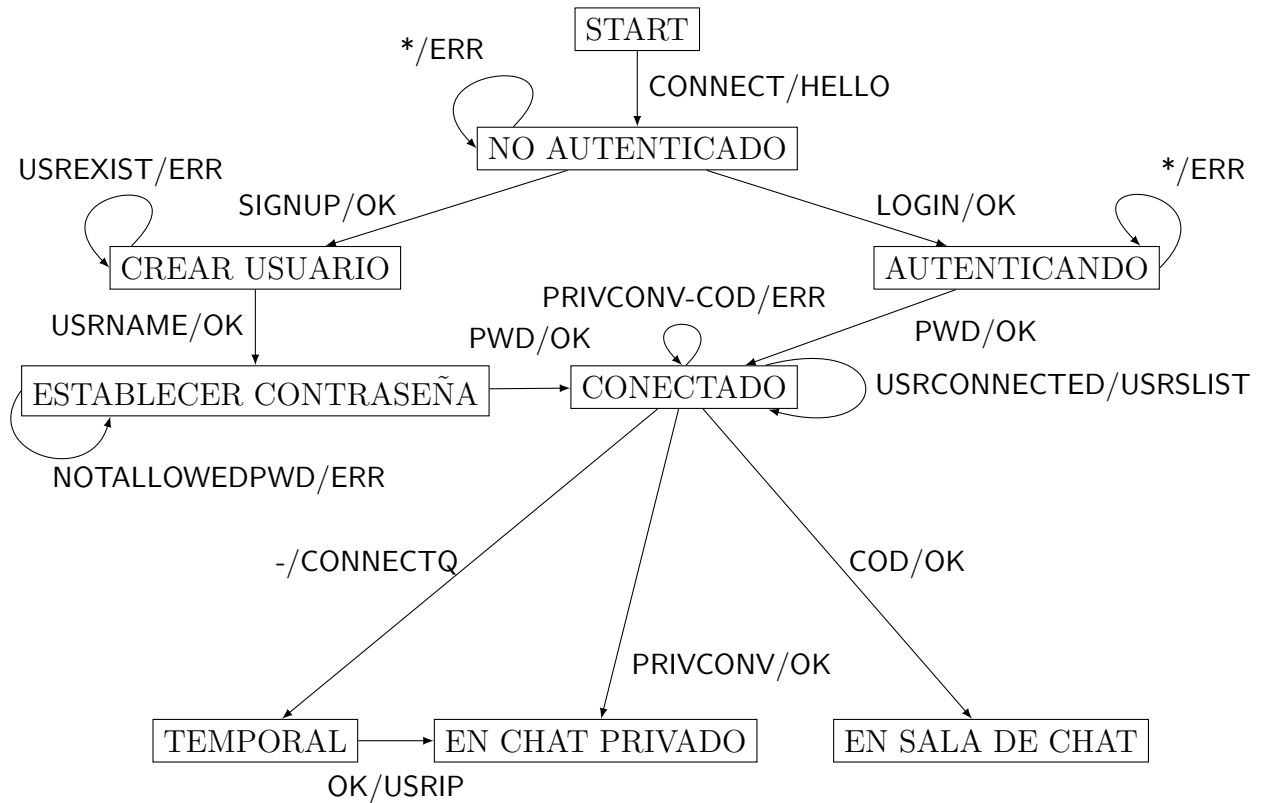
En esta práctica se va a desarrollar e implementar un protocolo de aplicación. En nuestro caso, vamos a desarrollar un pequeño servicio de mensajería instantánea para la terminal de linux, Tchat, que tendrá dos funcionalidades básicas. Por un lado, permitirá conectarse con otros usuarios directamente, para tener una conversación entre dos personas. Por otro, permitirá crear salas de chat en las que pueden entrar varios usuarios e interactuar entre sí. Además, tendremos una pantalla principal en la que tendremos los usuarios que están conectados en ese momento, así como las salas de chat global abiertas.

Para realizar esta conexión, tendremos un servidor que controlará los usuarios que hay conectados a la red (usando un servicio de directorio), y que permitirá que unos usuarios se conecten a otros. Además, será el encargado de gestionar el funcionamiento de las salas de chat, recibiendo los mensajes y enviándoselos a todos los usuarios conectados a dicha sala de chat.

La conexión entre dos usuarios, en cambio, será individual, es decir, el servidor enviará a ambos la dirección del otro, y se establecerá una conexión entre ellos, sin que los mensajes tengan que pasar por el servidor.

## 2 Diagrama de estados del servidor

Veamos ahora el diagrama de estados del servidor:



Para todo estado del diagrama, además de los mensajes que se observan, se tendría también un mensaje de desconexión (**LOGOUT/BYE**), que conectase cualquier estado con el estado (**END**). Se especifican aquí estos mensajes para que el diagrama no se sobrecargue.

### 3 Mensajes que intervienen

Vamos a ver ahora los mensajes que vamos a necesitar para el correcto funcionamiento de nuestra aplicación. Se hará una pequeña descripción de los mensajes que aparecen en el diagrama anterior.

Código	Cuerpo	Descripción
00	CONNECT	Mensaje enviado por el cliente para conectarse al servidor
01	HELLO	Mensaje de confirmación de conexión
02	<option>	Mensaje enviado por el cliente para entrar al servidor, bien para registrarse o para autenticarse
03	<usrname>	Mensaje con el nombre de usuario con el que el cliente quiere acceder al servidor
04	<pwd>	Mensaje con la contraseña del usuario
05	OK	Mensaje de confirmación
06	ERR	Mensaje de error del servidor
07	CONNECTQ + <userIP>	Mensaje en el que el servidor manda una petición de chat privado del usuario con IP <userIP>
08	<userIP>	Mensaje enviado por el servidor con la dirección IP del usuario a conectar
09	PRIVCONV + <user>	Mensaje enviado por el cliente para establecer un chat privado con el usuario <user>
10	COD <cod>	Mensaje con el código de una sala de chat
11	USRCONNECTED	Mensaje para solicitar la lista de usuarios conectados
12	<usrslst>	Mensaje con la lista de los usuarios conectados
13	LOGOUT	Mensaje de desconexión del cliente
14	BYE	Mensaje de confirmación de salida del servidor

### 4 Prototipo de la aplicación

Se va a desarrollar una pequeña aplicación que ilustre el funcionamiento básico de la aplicación. En particular, ilustra cómo se gestionaría la parte de chats entre dos usuarios. La aplicación desarrollada contará con un servidor que gestiona las conexiones, y dos tipos de clientes. Uno de ellos será un cliente escuchante, y otro un cliente emisor. El cliente escuchante podrá conectarse al servidor y especificar un nombre de usuario. Entonces, el servidor registrará su IP y el puerto en el que escucha en una base de datos, y le indicará que puede quedarse escuchando a que un cliente emisor solicite la conexión con él. Por el otro lado, el cliente emisor se conectará al servidor y solicitará la conexión a otro usuario utilizando su nombre de usuario. Si dicho usuario se encuentra conectado, el servidor responderá con la IP del escuchante y el puerto en el que se está escuchando. Entonces, el emisor se conectará a dicho usuario y empezará el envío de mensajes.

La idea final de la aplicación consistiría en tener en una misma ventana ambos tipos de clientes para poder tener una conversación bidireccional con otro usuario. Veamos ahora el funcionamiento del programa en distintas terminales. Primero, cuando se hace

la conexión del servidor:

```
paco@paco-K55VD ~/U/T/P/R/P/P/E/T/code (master)> java Server
Se ha lanzado correctamente el servidor de Tchat, esperando...
```

Ya tendríamos el servidor escuchando. Ahora, vamos a conectar un cliente escuchante al servidor, que lo va a registrar:

```
java /home/paco/Universidad/Tercero/PrimerCuatrimestre/Redes/Practicas/Practica2/Ejercicio5
paco@paco-K55VD ~/U/T/P/R/P/P/E/T/code (master)> java Server
Se ha lanzado correctamente el servidor de Tchat, esperando...
Se intentan conectar conmigo

java /home/paco/Universidad/Tercero/PrimerCuatrimestre/Redes/Practicas/Practica2/Ejercicio5
paco@paco-K55VD ~/U/T/P/R/P/P/E/T/code (master)> java ClientListener
Bienvenido a Tchat. Introduzca su nombre de usuario:
```

Ahora, introducimos el nombre de usuario, se nos registra correctamente, y se nos desconecta del servidor, esperando a recibir una nueva conexión por parte de un cliente emisor:

```

paco@paco-K55VD ~/U/T/P/R/P/P/E/T/code (master)> java Server
Se ha lanzado correctamente el servidor de Tchat, esperando...
Se intentan conectar conmigo
Cierro la conexión con el cliente
█

■ java /home/paco/Universidad/Tercero/PrimerCuatrimestre/Redes/Practicas/Practica2/Ejercicio5
paco@paco-K55VD ~/U/T/P/R/P/P/E/T/code (master)> java ClientListener
Bienvenido a Tchat. Introduzca su nombre de usuario: Paco
Registrado correctamente.
Esperando conexión entrante...
█

```

Se van a tener dos escuchantes registrados, para comprobar que la aplicación funciona correctamente:

```

paco@paco-K55VD ~/U/T/P/R/P/P/E/T/code (master)> java Server
Se ha lanzado correctamente el servidor de Tchat, esperando...
Se intentan conectar conmigo
Cierro la conexión con el cliente
Se intentan conectar conmigo
Cierro la conexión con el cliente
█

■ java /home/paco/Universidad/Tercero/PrimerCuatrimestre/Redes/Practicas/Practica2/Ejercicio5
paco@paco-K55VD ~/U/T/P/R/P/P/E/T/code (master)> java ClientListener
Bienvenido a Tchat. Introduzca su nombre de usuario: Paco
Registrado correctamente.
Esperando conexión entrante...
█

■ java /home/paco/Universidad/Tercero/PrimerCuatrimestre/Redes/Practicas/Practica2/Ejercicio5
paco@paco-K55VD ~/U/T/P/R/P/P/E/T/code (master)> java ClientListener
Bienvenido a Tchat. Introduzca su nombre de usuario: Mariadel
Registrado correctamente.
Esperando conexión entrante...
█

```

Ahora, vamos a conectar dos emisores, uno a cada uno de los receptores que tenemos conectados. Veamos la conexión de uno de ellos:

