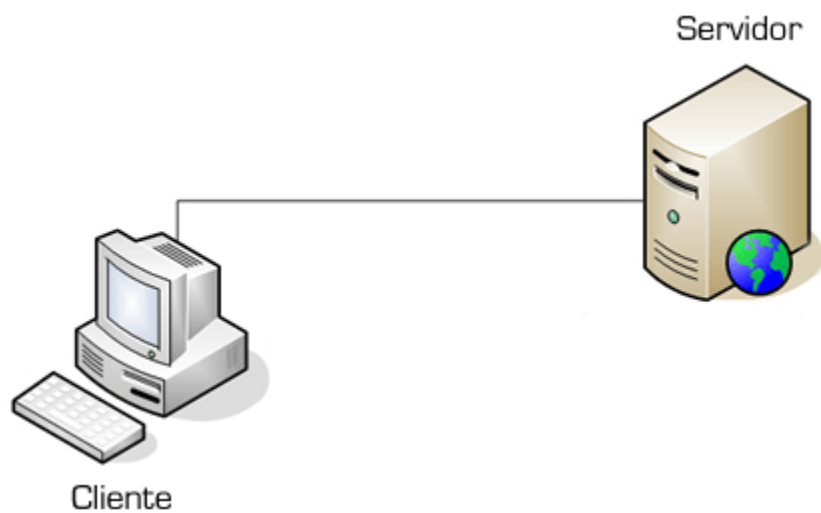


Práctica 3 TDRCII

Aplicación cliente/servidor

Profesor: Jorge Navarro Ortiz.



José Antonio Guerrero Avilés 75485683-M

Miguel León Ortiz 45869052-Z

INDICE

1.	Especificación del sistema.....	3
2.	Mensajes	3
3.	Diagramas.....	4
3.1	Diagramas de estados	4
3.2	Diagramas de flujo	6
4.	Implementación	10
4.1	Cliente	10
4.2	Servidor	10
5.	Pruebas de test.....	11
6.	Manual de usuario	14

1. Especificación del sistema

Vamos a realizar la interacción entre un cliente con un servidor en una aplicación de chat.

El cliente podrá darse de alta en el sistema e iniciar sesión en este. Una vez iniciada la sesión se puede enviar mensajes al servidor o pedirle que nos diga nuestra IP.

En nuestro caso es un modelo Cliente/Servidor Iterativo, lo que quiere decir que mientras un cliente esté conectado al servidor, no se podrá conectar ningún otro cliente, si no que este deberá esperar a que el primer cliente finalice su conexión para poder tener acceso al servidor.

2. Mensajes

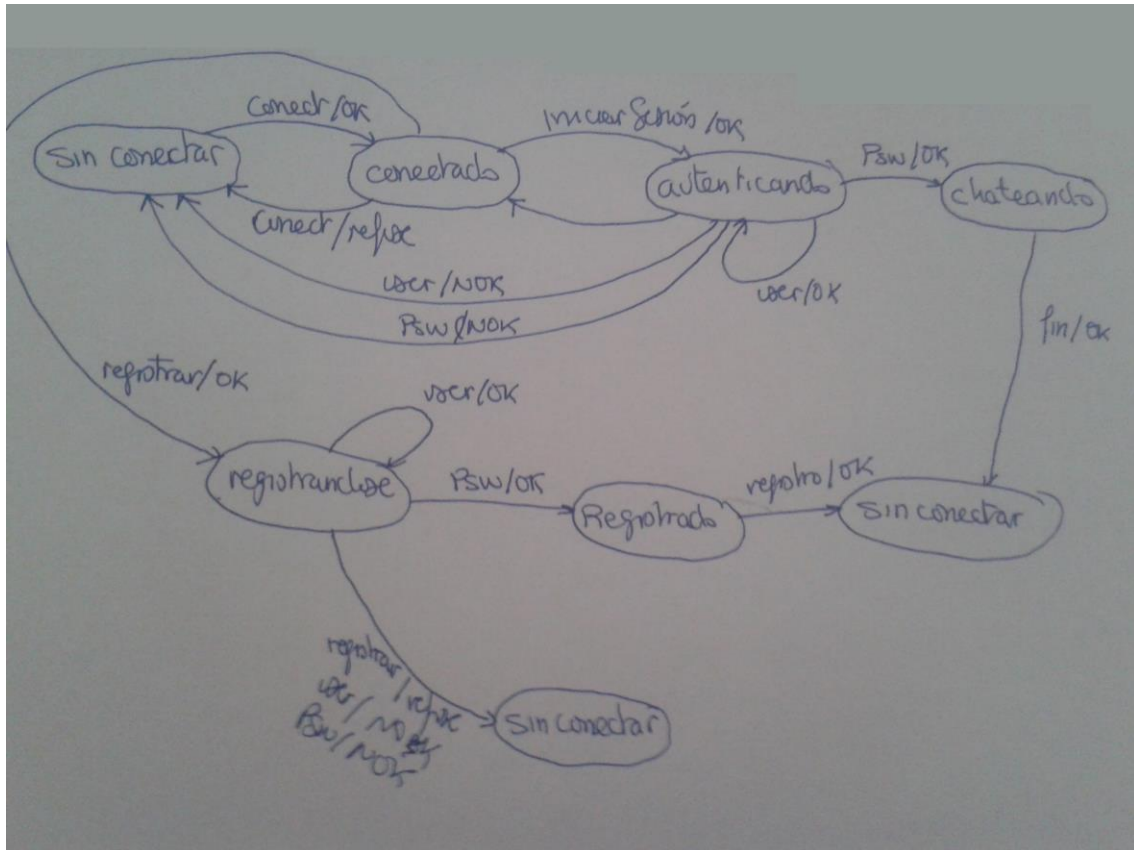
Mostrar Error:

- Fallo al enviar el usuario.
- Fallo al enviar el password.
- Usuario incorrecto.
- Password incorrecto.
- Alcanzado número máximo de intentos de inicio de sesión y/o intentos de registro.

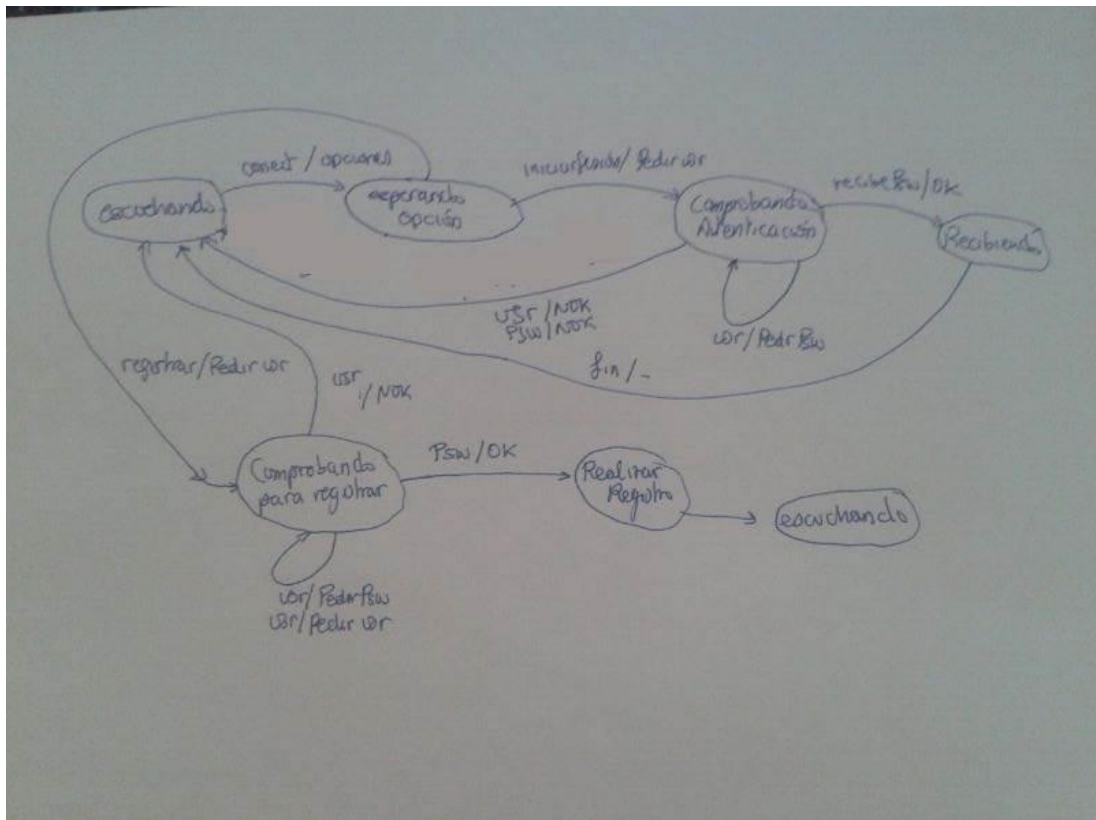
3. Diagramas

3.1 Diagramas de estados

Cliente:

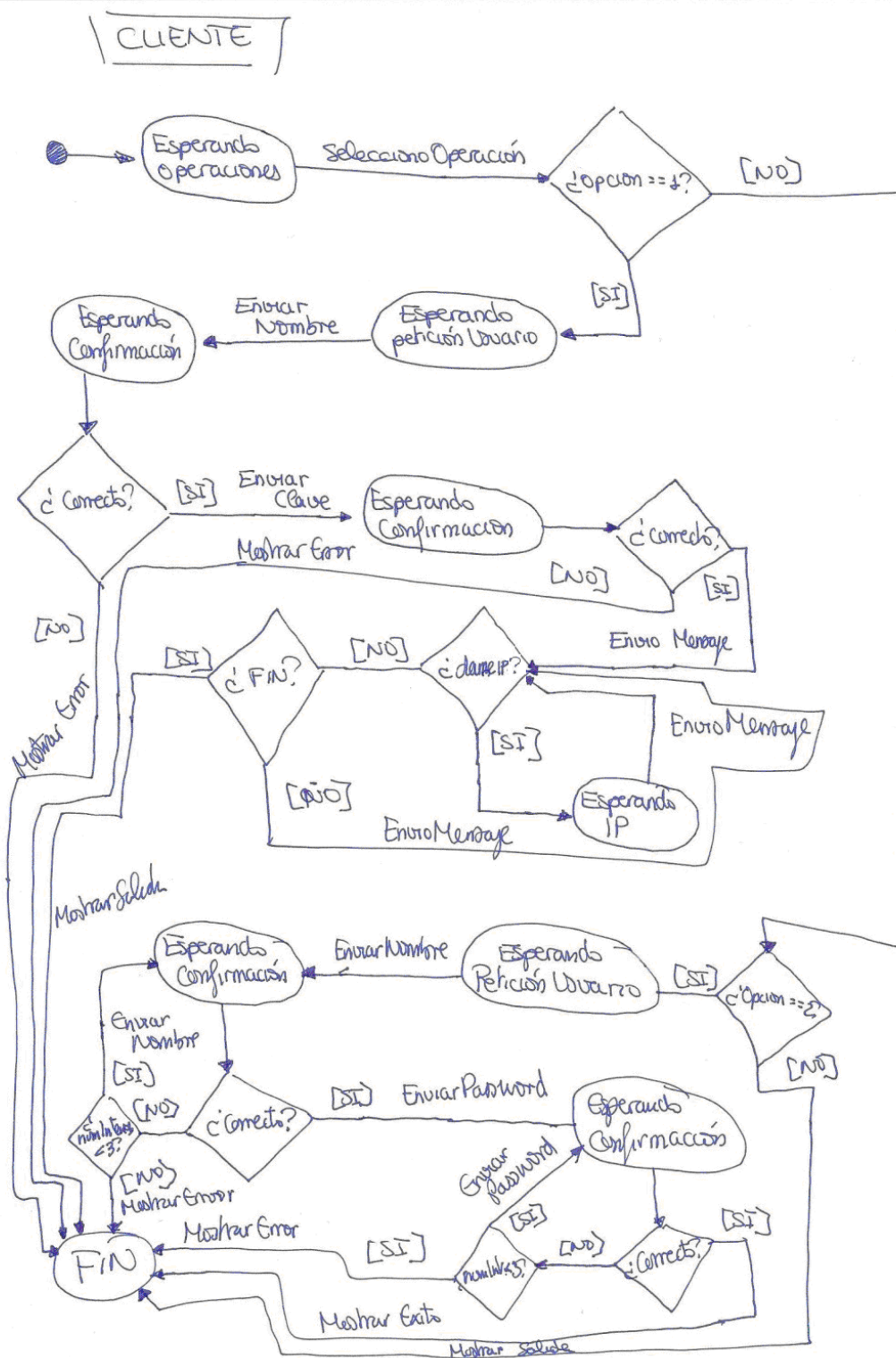


Servidor:



3.2 Diagramas de flujo

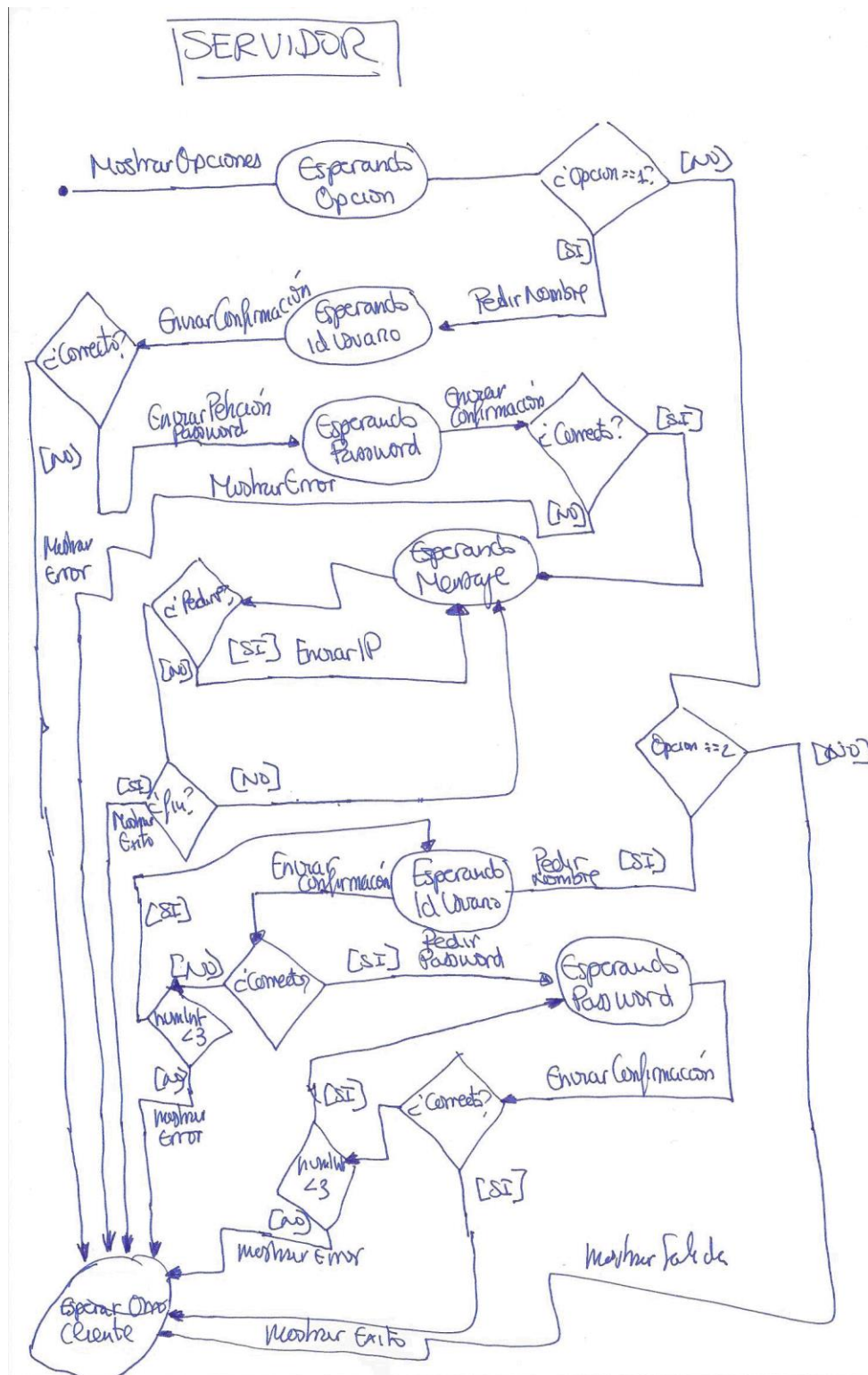
Cliente:



Operaciones utilizadas:

- seleccionarOperacion(idOperacion): El cliente tras ver las opciones posibles indicará qué acción desea realizar.
- mostrarError(mensajeError): Mostrará el mensaje de error ocurrido.
- EnviarNombre(nombre): El cliente enviará su nombre al servidor.
- EnviarClave(clave): El cliente enviará su clave al servidor.
- enviarMensaje(mensaje): Se enviará un mensaje al servidor

Servidor:



Operaciones utilizadas:

- `mostrarOpciones()`: Muestra las opciones que puede realizar el cliente
- `pedirNombre()`: El servidor le pide el nombre al cliente
- `pedirClave()`: El servidor le pide la clave al cliente.
- `enviarConfirmacion(confirmacion)`: El servidor le indica al cliente si los datos son correctos o no
- `enviarIP()`: El servidor envía la IP del cliente.

4. Implementación

Las características más relevantes de esta aplicación Cliente/Servidor iterativo, obviando el paso de mensajes que se dan para realizar la conexión entre ambos y que son propios de casi todas las aplicaciones Cliente/Servidor, son las siguientes:

4.1 Cliente

El cliente tiene un switch en el que hay dos opciones.

- 1.- representa el iniciar sesión en el sistema. Primero envía los datos y recibe las confirmaciones, contando hasta con 3 intentos. Cuando se consigue conectar envía mensajes al servidor.
- 2.- Representa el registrarse en el sistema. Se envían los datos correspondientes y se reciben las confirmaciones correspondientes.

4.2 Servidor

El servidor tiene un switch en el que hay dos opciones.

- 1.- representa el iniciar sesión por parte del cliente. Primero recibe los datos de usuario y password. Si se conecta correctamente este recibirá mensajes del cliente hasta que reciba 'fin'. Si el mensaje es 'dameip:', este enviara la ip del cliente
- 2.- Representa al cliente registrándose en el sistema. Se reciben los datos correspondientes y se envían las confirmaciones oportunas.

5. Pruebas de test

Entrada: Seleccionar iniciar sesión e introducir usuario erróneo en el primer intento.	Prueba 1
Salida Esperada: Error en el inicio de sesión. Usuario incorrecto.	
Salida Real: Indica que el usuario no existe y permite otra oportunidad.	
Prueba Correcta.	

Entrada: Seleccionar iniciar sesión e introducir usuario erróneo en el segundo intento.	Prueba 2
Salida Esperada: Error en el inicio de sesión. Usuario incorrecto.	
Salida Real: Indica que el usuario no existe y permite otra oportunidad.	
Prueba Correcta.	

Entrada: Seleccionar iniciar sesión e introducir usuario erróneo en el tercer intento.	Prueba 3
Salida Esperada: Error en el inicio de sesión y fin.	
Salida Real: Indica que el usuario no existe y finaliza la conexión.	
Prueba Correcta.	

Entrada: Seleccionar iniciar sesión e introducir usuario correcto.	Prueba 4.1
Salida Esperada: Se pide el password.	
Salida Real: Indica que el usuario es correcto y pide el password.	
Prueba Correcta.	

Entrada: Introducir password correcta tras la petición.	Prueba 4.2.1
Salida Esperada: Sesión iniciada.	
Salida Real: Sesión iniciada.	
Prueba Correcta.	

Entrada: Introducir password incorrecto en un primer intento tras la petición.	Prueba 4.2.2.1
Salida Esperada: Error en el inicio de sesión y permite otro intento.	
Salida Real: Permite un nuevo intento de password.	
Prueba Correcta.	

Entrada: Introducir password incorrecto en un segundo intento tras la petición.	Prueba 4.2.2.2
Salida Esperada: Error en el inicio de sesión y permite otro intento.	
Salida Real: Permite un nuevo intento de password.	
Prueba Correcta.	

Entrada: Introducir password incorrecta tras la petición.	Prueba 4.2.2.3
Salida Esperada: Error en el inicio de sesión y fin.	
Salida Real: Error y fin de sesión	
Prueba Correcta.	

Entrada: Seleccionar registrarse e introducir usuario existente.	Prueba 5
Salida Esperada: Error en el registro y fin.	
Salida Real: Indica que el usuario existe y finaliza la conexión.	
Prueba Correcta.	

Entrada: Seleccionar registrarse e introducir usuario no existente.	Prueba 6.1
Salida Esperada: Se pide password.	
Salida Real: Indica que el usuario es correcto y pide un password.	
Prueba Correcta.	

Entrada: Seleccionar registrarse e introducir usuario existente.	Prueba 6.2
Salida Esperada: Error en el registro y fin.	
Salida Real: Indica que el usuario existe y finaliza la conexión.	
Prueba Correcta.	

Entrada: Introducir el comando dameip: cuando estas chateando.	Prueba 7
Salida Esperada: Se obtiene la IP desde la que estas chateando.	
Salida Real: Se muestra la IP.	
Prueba Correcta.	

Entrada: Introducir el comando FIN	Prueba 8
Salida Esperada: Finaliza la conexión.	
Salida Real: Conexión finalizada.	
Prueba Correcta.	

Entrada: Escribir algo cuando chateas.	Prueba 9
Salida Esperada: Se muestra en el servidor.	
Salida Real: Texto mostrado en el servidor.	
Prueba Correcta.	

6. Manual de usuario

- Para compilar los programas se tendrá que introducir la orden make. (Con la orden make clean podemos eliminar los ejecutables y limpiar la compilación del proyecto))

- Para ejecutar el cliente se tendrá que poner lo siguiente:

./cliente num (Con num>=15000).

- Para ejecutar el servidor se tendrá que poner lo siguiente:

./servidor num (Con num igual que el que se ha puesto al cliente).

- Una vez iniciado cliente y servidor, en la pantalla del servidor aparecerá el mensaje “Escuchando...” y en la del cliente se mostrará un menú con 3 opciones:

1.- Iniciar Sesión.

2.- Registrarse.

0.- Salir.

Seleccionando una de estas opciones, el servidor irá enviando al cliente diferentes mensajes con menús y opciones o avisos que le permitirán realizar la operación que se haya seleccionado.