DOCUMENTACIÓN
PROYECTO NANOCHAT

Integrantes: Matías Nahuel Aguado Espíndola
Santiago Bernabé Leal

Grupo: 2

ÍNDICE

Formato de los mensajes y ejemplos……2

Autómatas de cliente, servidor y directorio……

Implementación del formato de los mensajes……

Mecanismo de gestión de salas…….

Mejoras adicionales implementadas……

Conclusiones…….

FORMATO DE LOS MENSAJES DEL DIRECTORIO

Formatos

Control

Opcode (1 byte)

OneParameter

Opcode (1 byte)	Parámetro (4 bytes)

TwoParameter

Opcode (1 byte)	Parámetro (1 byte)	Parámetro (4 bytes)

TwoParameter (variación)

Opcode (1 byte)	Parámetro (4 byte)	Parámetro (4 bytes)

Tipos de mensajes

Mensaje: buildRegistration (opcode = 1)

Formato: TwoParameter

Sentido de la comunicación: Cliente -> Directorio

Descripción: Este mensaje lo envía el Cliente al Directorio cuando quiere solicitar el registro a un servidor que viene dado con el Protocol_ID y el Port.

Ejemplo:

Opcode (1 byte)	Protocol_ID (1 byte)	Port (4 bytes)
1	6	16754

Mensaje: buildQuery (opcode = 2)

Formato: OneParameter

Sentido de la comunicación: Cliente -> Directorio

Descripción: Este mensaje lo envía el cliente al directorio cuando quiere consultar si existe algún servidor con el protocolo dado en Protocol.

Ejemplo:

Opcode (1 byte)	Protocol (4 bytes)
2	8

Mensaje: sendOK (opcode = 3)

Formato: Control

Sentido de la comunicación: Directorio -> Servidor

Descripción: Este mensaje lo envía el directorio al cliente para confirmar que el registro de este mismo al servidor.

Ejemplo:

Opcode (1 byte)
3

Mensaje: sendEmpty (opcode = 4)

Formato: Control

Sentido de la comunicación: Directorio -> Servidor

Descripción: Este mensaje lo envía el directorio al cliente para notificar que no se ha podido conectar con éxito al servidor pedido.

Ejemplo:

Opcode (1 byte)
4

Mensaje: sendServerInfo (opcode = 5)

Formato: TwoParameter

Sentido de la comunicación: Directorio -> Servidor

Descripción: Este mensaje lo envía el Directorio al cliente para entregar la dirección del servidor pedido por este.

Ejemplo:

Opcode (1 byte)	Direccion (4 bytes)	Puerto (4 bytes)
5	172.12.1.68	08953

Formato de los mensajes entre Servidor y Cliente

Formato de los mensajes (Lenguaje de Marcas)

NCRoomMessage

<message>

<operation>operation</operation>

<name>name</name>

</message>

NCSimpleMessage

<message>

<operation>operation</operation>

</message>

NCChatMessage

<message>

<operation>operation</operation>

<user>user</user>

<msg>msg</msg>

</message>

NCRoomListMessage

```
<message>
<operation>operation</operation>
        <rooms>
                <name>sala1</name>
                        <users>
                                <nick>user1</nick>
                                <nick>user2</nick>
                                ...
                        </users>
                <name>sala2</name>
                <name>sala3</name>
                ...
        </rooms>
</message>
```

Tipos de mensajes

Mensaje: Nick

Formato: NCRoomMessage

Sentido de la comunicación: Cliente -> Servidor

Descripción: Este mensaje es el usado para registrarse con un nick que establece el usuario.

Mensaje: Nick_ok

Formato: NCSimpleMessage

Sentido de la comunicación: Servidor -> Cliente

Descripción: Este mensaje se usa para confirmar que el nick que ha escogido es válido para el registro.

Mensaje: Nick_error

Formato: NCSimpleMessage

Sentido de la comunicación: Servidor -> Cliente

Descripción: Este mensaje se usa para responder al usuario que el nick que ha escogido no es válido (duplicado).

Mensaje: Get_roomlist

Formato: NCSimpleMessage

Sentido de la comunicación: Cliente -> Servidor

Descripción: Este mensaje es el que se utiliza para que el cliente solicite la lista de salas del servidor.

Mensaje: Roomlist

Formato: NCRoomListMessage

Sentido de la comunicación: Servidor -> Cliente

Descripción: Este mensaje es el usado para proporcionar al cliente la lista de usuarios antes pedida.

Mensaje: Enter

Formato: NCRoomMessage

Sentido de la comunicación: Cliente -> Servidor

Descripción: Este mensaje lo envía el cliente para solicitar entrar a una de las salas del servidor

Mensaje: Enter_ok

Formato: NCSimpleMessage

Sentido de la comunicación: Servidor -> Cliente

Descripción: Este mensaje es en respuesta a la solicitud de entrada a una sala, contestando que el acceso ha sido correcto.

Mensaje: Enter_error

Formato: NCSimpleMessage

Sentido de la comunicación: Servidor -> Cliente

Descripción: Este mensaje es en respuesta a la solicitud de entrada a una sala, contestando que el acceso no ha podido ser posible.

Mensaje: Get_info

Formato: NCRoomMessage

Sentido de la comunicación: Cliente -> Servidor

Descripción: Este mensaje se usa para solicitar información sobre la sala en la que se encuentra el cliente.

Mensaje: Info

Formato: NCRoomListMessage

Sentido de la comunicación: Servidor -> Cliente

Descripción: Este mensaje es en respuesta al anterior, que responde al cliente con la información de la sala pedida.

Mensaje: Send_message

Formato: NCRoomMessage

Sentido de la comunicación: Cliente -> Servidor

Descripción: Este mensaje es el que usa el cliente para enviar mensajes a través de una sala de chat.

Mensaje: Send_message_b

Formato: NCChatMessage

Sentido de la comunicación: Servidor -> Cliente

Descripción: Este mensaje es el que usa el servidor para enviar por broadcast el mensaje que un cliente ha enviado.

Mensaje: Exit

Formato: NCRoomMessage

Sentido de la comunicación: Cliente -> Servidor

Descripción: Este mensaje es el usado por el cliente para notificar la salida de la sala en la que se encuentra.

Mensaje: Rename

Formato: NCRoomMessage

Sentido de la comunicación: Cliente -> Servidor

Descripción: Este mensaje es el que se usa para que un cliente renombre la sala en la que se encuentra actualmente.

Mensaje: Rename_ok

Formato: NCSimpleMessage

Sentido de la comunicación: Servidor -> Cliente

Descripción: Este mensaje es el usado en respuesta al anterior, indicando al cliente que el renombre ha sido correcto.

Mensaje: Rename_error

Formato: NCSimpleMessage

Sentido de la comunicación: Servidor -> Cliente

Descripción: Este mensaje es el usado en respuesta al anterior, indicando al cliente que el renombre ha sido fallido.

Mensaje: Send_privmessage

Formato: NCChatMessage

Sentido de la comunicación: Cliente -> Servidor

Descripción: Este mensaje es el usado para que un cliente envíe un mensaje privado a otro de la misma sala en la que se encuentra.

Mensaje: Rcv_privmessage

Formato: NCSimpleMessage

Sentido de la comunicación: Servidor -> Cliente

Descripción: Este mensaje es el enviado por el servidor al cliente, indicando que el mensaje privado ha llegado bien.

Mensaje: New_room

Formato: NCRoomMessage

Sentido de la comunicación: Cliente -> Servidor

Descripción: Este mensaje es el que un cliente usa para solicitar crear una nueva sala en el servidor.

Mensaje: New_room_ok

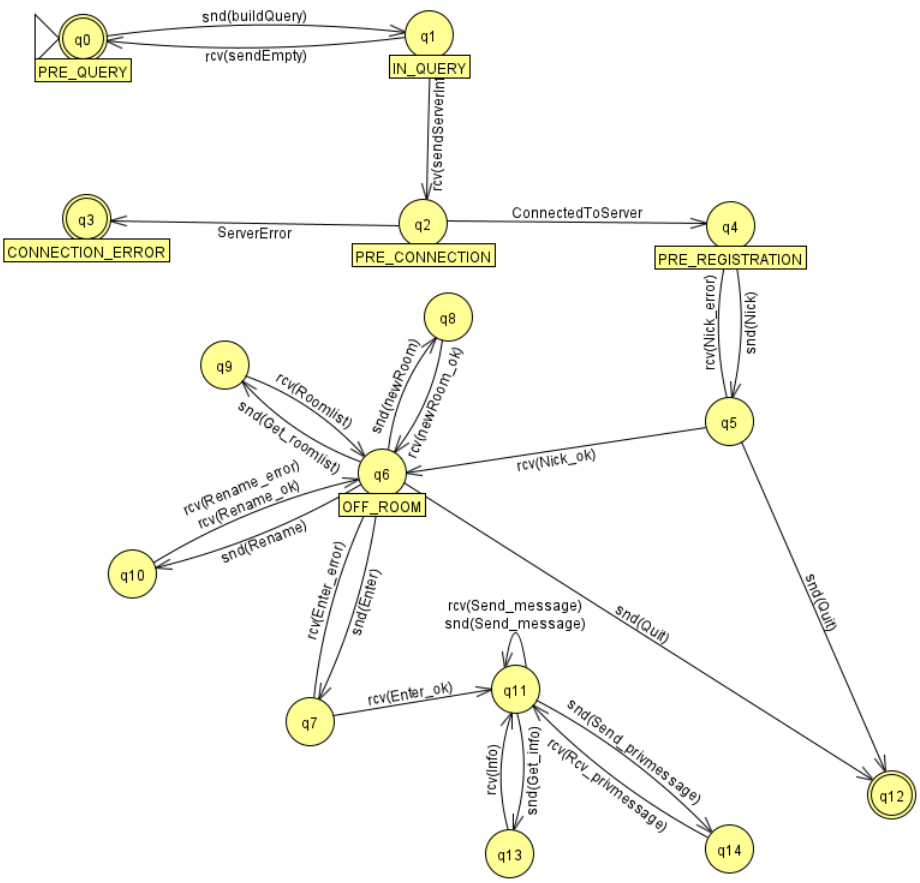
Formato: NCSimpleMessage

Sentido de la comunicación: Servidor -> Cliente

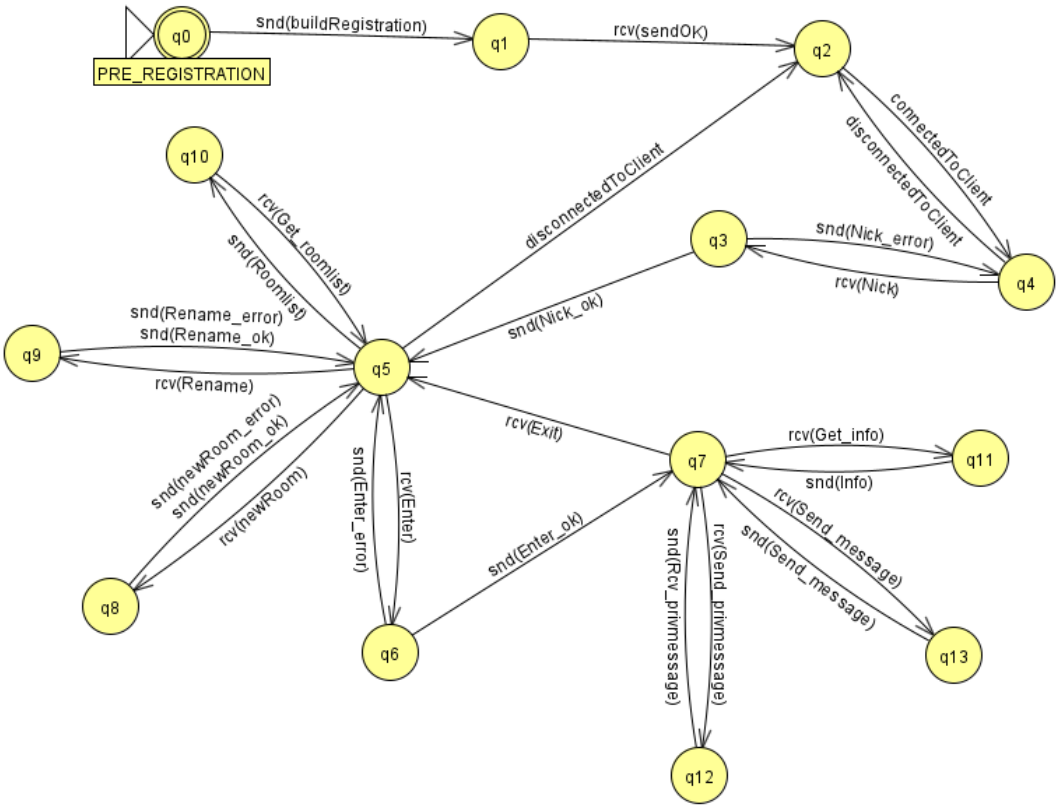
Descripción: Este mensaje se usa en respuesta al anterior, para indicar que la sala se ha creado con éxito.

AUTÓMATAS DE PROTOCOLO

Autómata de cliente



Autómata del servidor



MECANISMO DE GESTIÓN DE SALAS

Primeramente, el servidor crea por si mismo una sala inicial (RoomA). Además de esto, un cliente cuando esta conectado a este, es capaz de crear más salas como esta haciendo uso del comando newRoom. Puede crear cuántas salas quiera. Estas salas creadas tendrán las mismas funcionalidades que la sala que el servidor crea inicialmente. Por último, cabe destacar que cualquier cliente puede modificar el nombre de cualquier sala, aunque no la haya creado él, solo haciendo uso del comando rename, siempre y cuando se encuentre dentro de la sala en cuestión.

MEJORAS ADICIONALES IMPLEMENTADAS

En cuanto a las mejoras adicionales implementadas, hay que destacar las 3 obligatorias en la convocatoria de julio, como son: notificar a los demás usuarios de la sala las entradas/salidas de otros usuarios en las salas conforme se vayan produciendo; la posibilidad de crear nuevas salas en el servidor; y enviar un mensaje privado a un usuario de la sala.

Además de estas, hemos implementado la opción de renombrar las salas del servidor. Esto se ha hecho de una manera muy sencilla a través de las funciones renameRoom de NCConnector y NCController, dentro de la parte de client.

CONCLUSIONES

Como conclusión, hemos de decir que esta práctica se nos ha hecho algo pesada en un principio, pero conforme hemos ido haciéndola poco a poco, empezó a ser más llevadera e interesante. En cuando al grado de dificultad, la mayoría de funciones no nos ha supuesto un reto muy grande, aunque cabe destacar que el hecho de implementar los mensajes a través de Lenguaje de Marcas si nos pareció algo difícil en un principio.