

Instituto Politécnico Nacional

Escuela Superior de Cómputo

*Aplicaciones para comunicaciones de red*

*“Practica 3 – Chat”*

**Grupo:** 3CM15

**Integrantes:**

* Ramírez Olvera Guillermo.
* Sánchez Méndez Edmundo Josue.

**Fecha de entrega:** 7 de mayo de 2021

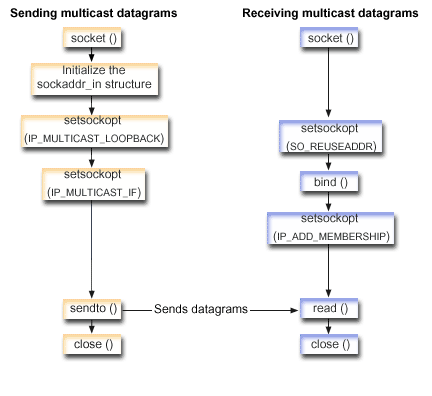
**Profesor:** Moreno Cervantes Axel Ernesto



A veces nos interesa que un ordenador pueda enviar un mensaje por red y que este sea recibido por otros ordenadores simultáneamente. Para ello están las direcciones multicast.

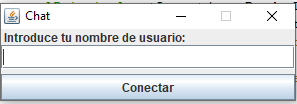
Son direcciones en el rango 224.0.0.0 a 239.255.255.255. La 224.0.0.0 está reservada y no puede usarse. Enviando mensajes por estas direcciones, cualquier otro ordenador en la red que las escuche podría leer dicho mensaje, independientemente de cuál sea la IP real de ese ordenador.

Es decir, si un ordenador quiere enviar un mensaje simultáneamente a varios, puede hacerlo enviando el mensaje a una de estas IPs, los demás ordenadores deben estar a la escucha de dichas IPs para recibir el mensaje.

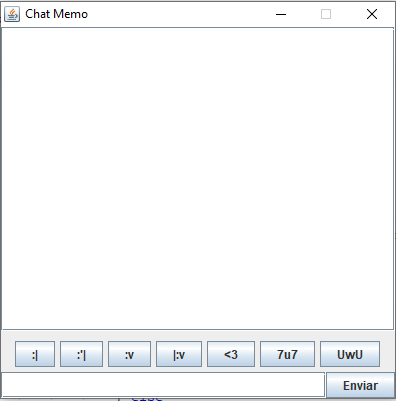


**Desarrollo.**

Para el desarrollo de esta práctica se generó los siguiente, primero una interfaz donde se coloca el nombre del usuario, mientras que la segunda es el chat principal.



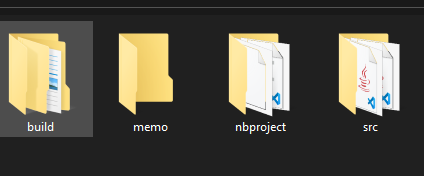
Interfaz de introducción de nombre.



Interfaz del chat.

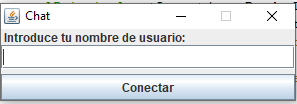
Una vez dado esto el usuario está en posición de entablar conversaciones en la sala, mandar y recibir archivos, y por último mandar mensajes privados.

Una consideración más es, debido al control de los usuarios el programa tiene la facultad de crear carpetas de usuario de acuerdo con el nombre de este.

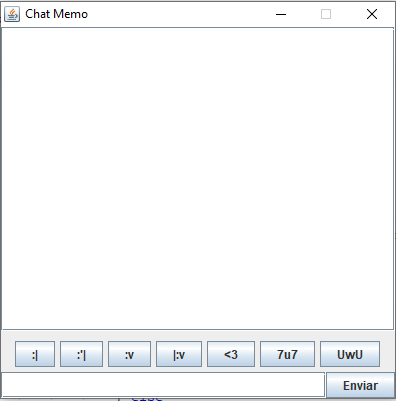


Carpeta Creada para el usuario **Memo**.

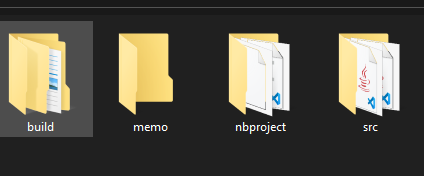
Pruebas.



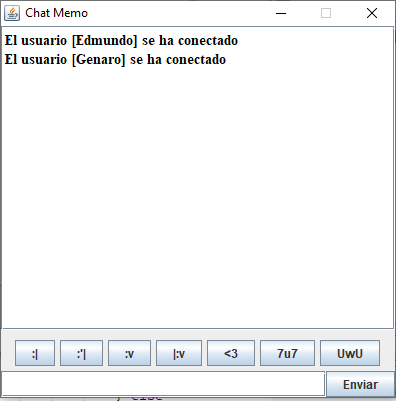
Usuario Registrado.



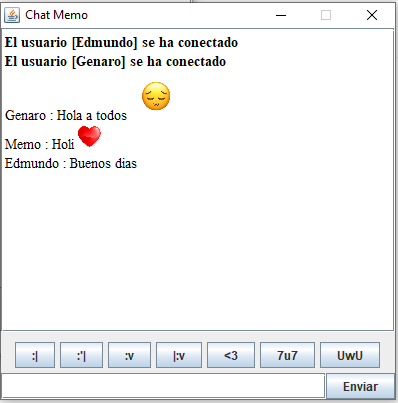
Interfaz de usuario principal.



Carpeta de usuario creada.



Anuncio de nuevos usuarios.



Envió de mensajes.

Dificultades Encontradas.

El primer punto para superar fue el comprender de buena manera como funcionaban los sockets multicast, ya que en un principio se tenía la vaga idea de usar un cliente y servidor, pero eso no era fácil de usar.

Una vez pasado este punto el siguiente, fue como actualizar la lista de contactos ya que en un punto pasaba de que se agregaban a modo que llegaban, es decir el primer usuario registrado es el que tenía la lista completa de todos los que se conectaban, el truco estuvo en poner un segundo anuncio que mandaba el nombre de usuario ya existente al recibir uno nuevo.

Otro punto para resaltar fue el hecho de sobrecargar el constructor del objeto Mensaje, con el propósito de poder mandar mensajes y archivos sin problemas entre el envío de mensajes y archivos.

Por último, el uso de hilos fue imprescindible para evitar que la interfaz se quedara a la espera del final del proceso para poder seguir funcionando, el uso de clases internas fue esencial para poder usar diversos hilos con diferentes usos.

Como posibles mejoras existen demasiadas, como el hecho de hacer anuncios de los usuarios ya conectados aparte de añadirlos a la lista de contactos, validaciones de cuando el usuario ya está conectado no se pueda registrar otro con el mismo nombre y quitar los usuarios de la lista de contactos cuando se desconecta.

Conclusiones.

En esta práctica pudimos observar las diferencias entre los sockets multicas y los de flujo. Pero la versatilidad que nos dan es amplia para hacer aplicaciones que deben tratar varios elementos interactuando entre sí, pero cuidando la integridad de los datos que se van a mandar, puesto que con UDP hay que buscar la forma de garantizar el envío de paquetes y el uso de hilos para soportar la concurrencia de múltiples usuarios al mismo tiempo.

Para los emojis hubo que cambiar algunos caracteres del texto del botón, ya que el analizador de expresiones regulares de Java devolvía errores al usar paréntesis o llaves.