

MÓDULO:

Programación de Servicios y Procesos

Tarea No Evaluable

Tarea 3

Chat Multicast
EJERCICIO 3



Unión Europea

Fondo Social Europeo
"El FSE invierte en tu futuro"

SOCKET UDP EJERCICIO 3 Chat Multicast

EJERCICIO

Realizar un servidor Multicast de forma que cada vez que se escriba por teclado un mensaje en dicho servidor, se encargará de enviarlo a los clientes multicast que se hayan unido al grupo. El proceso finalizará cuando el usuario escriba el mensaje “salir” debiendo de enviar el correspondiente mensaje a los clientes y a continuación cerrarse tanto el servidor como cada uno de los clientes.

REALIZACION

Crear una clase **MultiCastServer** que nada más ejecutarse muestre el siguiente mensaje:

“Arrancando el servidor multicast...”

Crear a continuación un socket Multicast y un objeto de tipo InetAddress llamado **grupo** que corresponderá al grupo multicast sobre la ip de clase D 231.0.0.1

```
InetAddress group = InetAddress.getByName("231.0.0.1");
```

Crear un objeto BufferedReader para leer por teclado.

En un bucle (cuya condición de salida será que lo escrito por teclado sea el texto “salir”):

- leer la entrada por teclado.
- Convertir dicho texto introducido por teclado en bytes para enviar en un datagrama
- Crear un DatagramPacket para enviar el texto introducido a los clientes del grupo Multicast, que se conecten a través del puerto 10000
- Proceder al envío de dicho datagrama

La finalización de la clase MultiCastServer será cerrando el socket

Crear una clase **MultiCastCliente** que se encargará de crear un socket Multicast a través del puerto 10000

Configurar el grupo al que nos vamos a conectar

Procederemos a unir al cliente al grupo Multicast

En un bucle (cuya condición de salida será que lo recibido desde el servidor sea el texto “salir”):

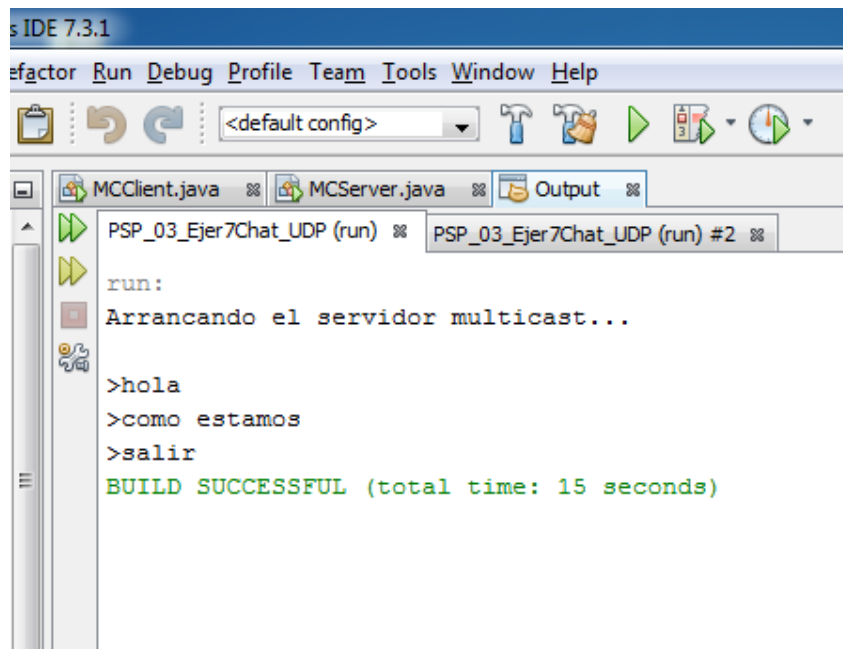
- Crear un datagrama para recibir los paquetes
- Recibir el datagrama que envía el Servidor Multicast
- Convertir la cadena de bytes que trae el datagrama en un String, limpiando los espacios en blanco que se utilizan para completar el tamaño del paquete
mensaje = (new String(buffer)).trim(); //donde buffer es el array de bytes del datagrama
- Mostrar la cadena en la pantalla de ejecución del cliente.
- El Cliente Multicast finalizará cuando el mensaje enviado por el Servidor (limpio y sin espacios de relleno en blanco) sea **“salir”**

Si el servidor envía salir, habrá que abandonar el grupo y cerrar el socket

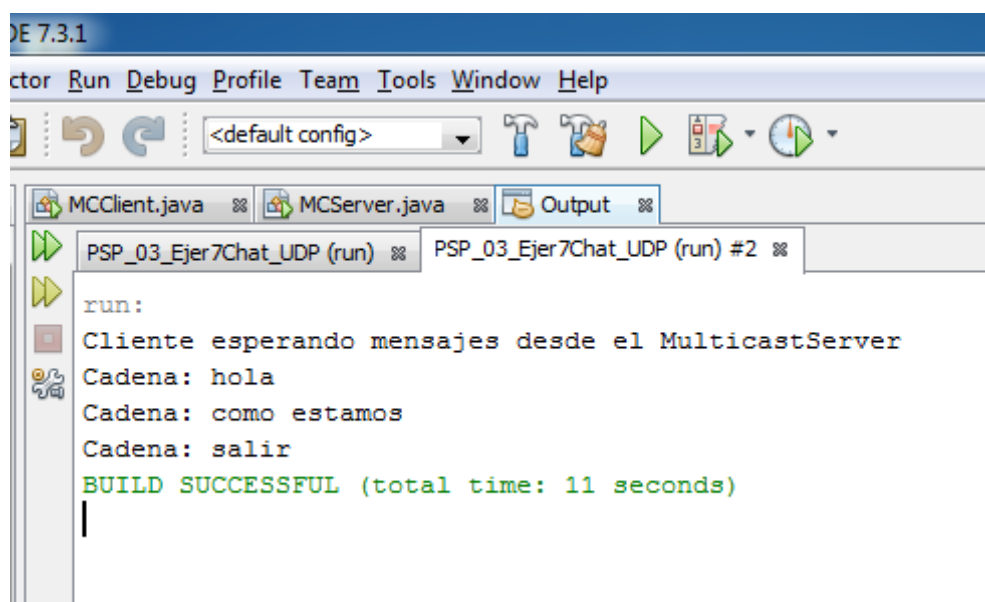
EJECUCIÓN

Ejecutar el MultiCastServer y a continuación una vez arrancado, ejecutar 1, 2 o más MultiCastCliente, escribiendo en el Server un mensaje y debiendo de recibirse dicho mensaje en cada uno de los clientes que se han unido al grupo.

Para finalizar escribir "salir" en el Server y se deberían de cerrar la ejecución tanto en el server como en los cliente.



```
IDE 7.3.1
efactor Run Debug Profile Team Tools Window Help
<default config>
MCClient.java MCServer.java Output
PSP_03_Ejer7Chat_UDP (run) PSP_03_Ejer7Chat_UDP (run) #2
run:
Arrancando el servidor multicast...
>hola
>como estamos
>salir
BUILD SUCCESSFUL (total time: 15 seconds)
```



```
IDE 7.3.1
efactor Run Debug Profile Team Tools Window Help
<default config>
MCClient.java MCServer.java Output
PSP_03_Ejer7Chat_UDP (run) PSP_03_Ejer7Chat_UDP (run) #2
run:
Cliente esperando mensajes desde el MulticastServer
Cadena: hola
Cadena: como estamos
Cadena: salir
BUILD SUCCESSFUL (total time: 11 seconds)
```