

MÓDULO:

Programación de Servicios y Procesos

**Tarea No Evaluable**

**Tarea 4**

**Chat Multicast con Hilos**  
**EJERCICIO 4**



**Unión Europea**

Fondo Social Europeo  
*"El FSE invierte en tu futuro"*

## SOCKET UDP EJERCICIO 4 Chat Multicast con hilos

### EJERCICIO

Hacer una clase ChatMulticast que se encargará de funcionar como Emisor y Receptor en un chat Multicast. Para ello el ServidorEmisor creará un socketMulticast y se reunirá al grupo para enviar mensajes; esta clase cada vez que se ejecuta lanzará un hilo ChatThread que se encargará de realizar el papel de ClienteReceptor para recibir mensajes y visualizarlos.

### REALIZACION

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

**“Uniendose al chat multicast...”**

Crear a continuación un socket Multicast en el puerto 10000 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");`

Una vez que se ha creado el socket y nos hemos unido al grupo, la clase lanzará un hilo que hará de receptor para dicho usuario **ChatThread**, pasando como parámetro el socket

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 se producirá cuando el usuario teclee “salir” procediendo a dejar el grupo y cerrar el socket. En ese momento el usuario dejará de estar conectado al chat, quedando el resto de los usuarios conectados.

Crear una clase **ChatThread** que extenderá de Thread y en cuyo constructor recibirá como parámetro el socket.

El run del hilo será 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 del Servidor
- 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.

Simular la ejecución de varios usuarios lanzando varias veces **MultiCastServer** y probar a escribir un mensaje en cualquiera de las ventanas de ejecución y si lo reciben los demás. Probar a finalizar alguna sesión de usuario y comprobar si abandona el chat.

```

run:
Reunida al Grupo Multicast: 231.0.0.1 Puerto:10000
Escribe cualquier cosa y RETURN, o salir para finalizar
>hola
mensajehola Longitud: 4
>

Recibido: "hola"
Mensaje Longitud: 4

```

```

run:
Reunida al Grupo Multicast: 231.0.0.1 Puerto:10000
Escribe cualquier cosa y RETURN, o salir para finalizar
>

Recibido: "hola"
Mensaje Longitud: 4
>mensaje enviado desde el chat4
mensaje>mensaje enviado desde el chat4 Longitud: 31
>

Recibido: ">mensaje enviado desde el chat4"
Mensaje Longitud: 31

```

```

run:
Reunida al Grupo Multicast: 231.0.0.1 Puerto:10000
Escribe cualquier cosa y RETURN, o salir para finalizar
>hola
mensajehola Longitud: 4
>

Recibido: "hola"
Mensaje Longitud: 4

Recibido: ">mensaje enviado desde el chat4"
Mensaje Longitud: 31
salir
mensajesalir Longitud: 5
Dejando el grupo...
Cerrando el hilo
BUILD SUCCESSFUL (total time: 2 minutes 0 seconds)

```