****

PRÁCTICA FINAL

**IMPLEMENTACIÓN DE UN CHAT MULTICAST**

**SISTEMAS DISTRIBUIDOS 4ºGITT**

En esta práctica se ha desarrollado un chat multicast. Para ello, se ha creado una interfaz gráfica.

Los mensajes multicast son mensajes que se envían a distintos puntos simultáneamente. Se definen unas direcciones IP reservadas para este tipo de mensajes, las cuales pertenecen al siguiente rango [224.0.0.0 a 239.255.255.255]. Un host que se une a una de estas direcciones recibe todos los mensajes procedentes del grupo.

El código se ha basado en un fichero escrito en Python. El cual hace uso de una clase con distintas funciones. A continuación, se explicará el código y la implementación de las distintas funciones.

Para empezar, se importan las librerías necesarias para el desarrollo del código

A continuación, se define la clase Aplicación, esta será la encargada de definir la interfaz gráfica, en la cual se encuentra toda la lógica de la aplicación. Dicha clase, cuenta con distintos métodos.

El primero, llamado \_\_init\_\_, hace referencia a la inicialización de la clase, los parámetros que recibe son la dirección ip de grupo multicast, el puerto y la dirección de enlace. Todos ellos vienen definidos por defecto, en caso de que no se indique ningún valor de entrada.

A continuación, observamos el desarrollo del código:

Captura de pantalla de un celular con letras

Descripción generada automáticamente

Captura de pantalla de un celular

Descripción generada automáticamente

La siguiente función de la clase que nos encontramos es onEnter, dicha función únicamente llama a la función enviar. Se ha definido para poder enviar los mensajes con el teclado.

La función enviar se encarga de recoger el mensaje escrito, pasarlo a la función actualizar y limpiar la caja de texto.

En la función actualizar se crea una hebra, para poder recibir los mensajes mientras se desarrolla la aplicación. Además, enviamos el mensaje al grupo y actualizamos todos los mensajes llamando a la función actualizoLabel.

La función clientThread, implementa el desarrollo de la hebra. Se crea un bucle infinito para recibir todos los mensajes del grupo en tiempo real. Una vez recibido se actualizan todos los mensajes llamando a la función actualizoLabel

Finalmente, la función actualizoLabel, simplemente escribe en la interfaz todos los mensajes de los usuarios recibidos hasta el momento.

Una captura de pantalla de una computadora

Descripción generada automáticamente

Finalmente, nos encontramos con la función main, la cual es la principal, únicamente llama a la clase.

Imagen que contiene captura de pantalla, dibujo

Descripción generada automáticamente

Un ejemplo del desarrollo de la práctica es el siguiente, desde la terminal en la línea de comandos ejecutamos el código.

Primero se nos solicitará que introduzcamos un usuario el cual identificará nuestros mensajes, una vez escrito, aparecerá la interfaz gráfica.

Para ver un ejemplo de conversación entre dos usuarios, abrimos otra ventana de la terminal y volvemos a ejecutar el código.

Captura de pantalla de un celular

Descripción generada automáticamente