**ESCUELA POLITÉCNICA NACIONAL**

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**APLICACIONES DISTRIBUIDAS (TDSD523)**

|  |  |
| --- | --- |
|  | |
| ASIGNATURA: | Aplicaciones Distribuidas |
| PROFESOR: | Ing. Vanessa Guevara. |
| PERÍODO ACADÉMICO: | 2023-B |
|  | |

**Trabajo en Clase**

|  |  |
| --- | --- |
|  | |
| TÍTULO:  **Implementación de Hilos y Sockets con UDP en Java** | |
|  | |
|  | |
| UAI Noticias - El Portal de Noticias de la Universidad |  |
|  | |

# OBJETIVOS

* Comprender los conceptos básicos de los sockets e hilos.
* Crear un servidor de chat y uno o varios clientes de chat utilizando sockets UDP e hilos en Java.
* Implementar hilos en Java para permitir una comunicación concurrente y paralela entre el servidor y los clientes.

# DESARROLLO Y RESULTADOS DE LA PRÁCTICA

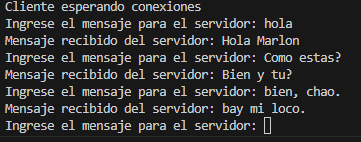
1. SERVIDOR

Aquí detallar y explicar los procesos realizados para implementar el servidor (como un manual de implementación). Incluir capturas de pantalla

1. CLIENTE

* Importa las clases de DatagramSocket, DatagramPacket e InetAddress para crear el socket y levantar el servidor UDP en Java.
* Se ingresa la dirección IP del servidor al que se quiere conectar, usando la clase InetAddress. Esta clase representa una dirección IP, que puede ser una cadena de texto o un arreglo de bytes.
* Se crea un bucle principal para mantener la comunicación con el servidor.
* Se usa la clase Scanner para leer un mensaje desde la terminal del cliente. Esta clase te permite leer datos de diferentes tipos desde la entrada estándar
* se crea un arreglo de bytes para almacenar el mensaje que se quiere enviar al servidor. Se debe convertir el mensaje de tipo String a tipo byte[], usando el método getBytes().
* Se crea un objeto de tipo DatagramPacket, que representa el paquete de datos que se va a enviar al servidor.
* Se crea otro objeto de tipo DatagramSocket, que se encarga de enviar el paquete al servidor. Se debe usar el método send() para enviar el paquete.
* Se crea otro arreglo de bytes para recibir el mensaje que envía el servidor. Se debe especificar una longitud suficiente para almacenar el mensaje
* Se crea otro objeto de tipo DatagramPacket, que representa el paquete de datos que se va a recibir del servidor.
* se usa el método receive() del socket del cliente para recibir el paquete del servidor.
* Se extrae el mensaje del paquete recibido, usando el método getData() para obtener el arreglo de bytes con el mensaje, el método getLength() para obtener la longitud del mensaje
* Se muestra el mensaje recibido del servidor en la terminal del cliente, usando el método println() de la clase System.

1. DEMOSTRACION



# CONCLUSIONES

(3 Conclusiones)

# BIBLIOGRAFÍA

**PRESENTACIÓN**

Una vez culminada tu tarea, súbela en este apartado.

Recuerda el nombre del archivo deberá ser: **TC\_APDIST\_2023B\_NApellido**.