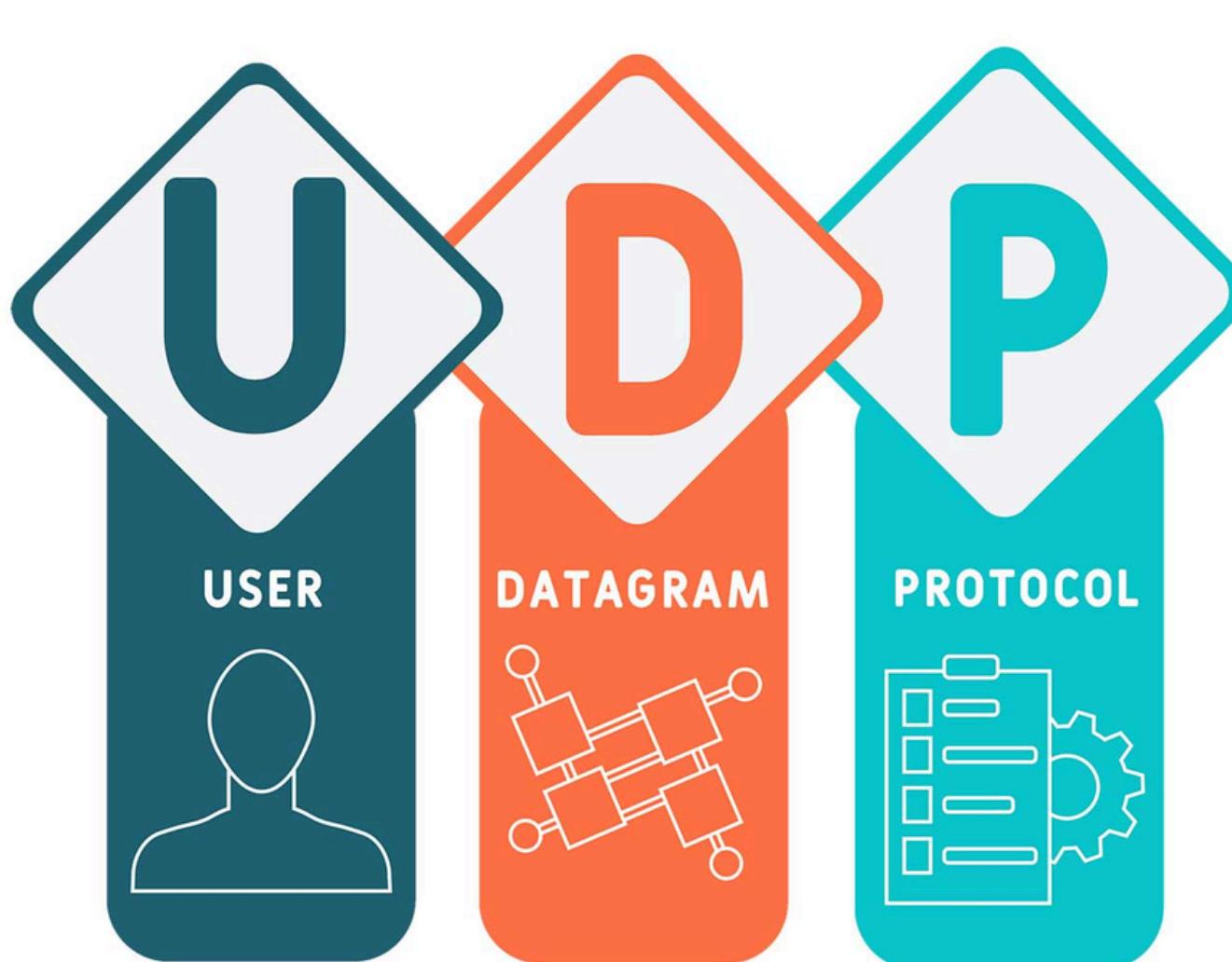


CLIENTE SERVIDOR

BETZY AVENDAÑO MARTINEZ

QUE ES UDP?

ES UN PROTOCOLO DE COMUNICACIÓN DE INTERNET QUE SE CARACTERIZA POR SER RÁPIDO Y EFICIENTE PORQUE NO ESTABLECE UNA CONEXIÓN FORMAL ANTES DE ENVIAR DATOS



IMPORTACIONES DE LIBRERIAS

PERMITEN USAR ENTRADA/SALIDA DE DATOS, COMUNICACIÓN DE RED, INTERFAZ GRÁFICA Y DISEÑO VISUAL

```
....  
package cliente;  
import java.io.*;  
import java.net.*;  
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
import java.text.SimpleDateFormat;  
import java.util.Date;
```

MÉTODO ENVIARMENSAJE

Crea una conexión con el servidor, envía el mensaje y espera respuesta

Componentes:

- Socket: Punto de comunicación entre cliente y servidor
- hostActual: Dirección del servidor
- puertoActual: Número de puerto donde el servidor escucha

```
PRIVATE VOID ENVIARMENSAJETCP(STRING MENSAJE)
    THROWS IOEXCEPTION {
        SOCKET SOCKET = NULL;
        TRY {
            SOCKET = NEW SOCKET(HOSTACTUAL,
                PUERTOACTUAL);
            SOCKET.SETSOFTIMEOUT(5000);

            PRINTWRITER SALIDA = NEW
            PRINTWRITER(SOCKET.GETOUTPUTSTREAM(), TRUE),
            SALIDA.PRINTLN(MENSAJE);

            BUFFEREDREADER ENTRADA = NEW
            BUFFEREDREADER(
                NEW
                INPUTSTREAMREADER(SOCKET.GETINPUTSTREAM())
            );
            STRING RESPUESTA = ENTRADA.READLINE();

            IF (RESPUESTA != NULL) {
                SWINGUTILITIES.INVOKEELATER(() -> {
                    AGREGARMENSAJESERVIDOR(RESPUESTA);
                });
            } ELSE {
                THROW NEW IOEXCEPTION("EL SERVIDOR CERRÓ
LA CONEXIÓN SIN ENVIAR RESPUESTA");
            }
        }
    }
}
```

MÉTODO MAIN()

```
PUBLIC STATIC VOID MAIN(STRING[] ARGS) {  
  
    TRY {  
  
        UIMANAGER.SETLOOKANDFEEL(UIMANAGER.GET  
            SYSTEMLOOKANDFEELCLASSNAME());  
    } CATCH (EXCEPTION E) {  
        E.PRINTSTACKTRACE();  
    }  
  
    SWINGUTILITIES.INVOKELATER(() ->  
        CLIENTE CLIENTE = NEW CLIENTE();  
        CLIENTE.SETVISIBLE(TRUE);  
    )
```

Es el punto de entrada de cualquier aplicación Java. Es lo primero que se ejecuta cuando inicias el programa.

SERVIDOR

```
public class servidor {  
  
    public static void main(String[] args) {  
        // Escoge UNO de los dos servidores:  
  
        ...  
        servidorTCP();      // <- Descomentar para usar  
        // servidorUDP();   // <- Descomentar para usar  
    }  
}
```

- **TEXPROGRAMA QUE FUNCIONA COMO SERVIDOR DE MENSAJERÍA**
- **PUEDE TRABAJAR EN MODO TCP O MODO UDP**
- **RECIBE MENSAJES DE CLIENTES Y ENVÍA RESPUESTAS**
- FUNCIÓN PRINCIPAL:**
- **ESCUCHA EN UN**

FINAL INT PUERTO = 3306;

```
TRY {
    SERVERSOCKET SERVIDOR = NEW
    SERVERSOCKET(PUERTO);
    SYSTEM.OUT.PRINTLN("SERVIDOR TCP INICIADO
EN PUERTO " + PUERTO);

    WHILE (TRUE) {
        // ESPERAR CLIENTE
        SOCKET SOCKET = SERVIDOR.ACCEPT();
        SYSTEM.OUT.PRINTLN("CLIENTE
CONECTADO");

        // LEER MENSAJE DEL CLIENTE
        BUFFEREDREADER ENTRADA = NEW
        BUFFEREDREADER(
            NEW
        INPUTSTREAMREADER(SOCKET.GETINPUTSTREAM())
        );
        STRING MENSAJE = ENTRADA.READLINE();
        SYSTEM.OUT.PRINTLN("CLIENTE DICE: " +
MENSAJE);

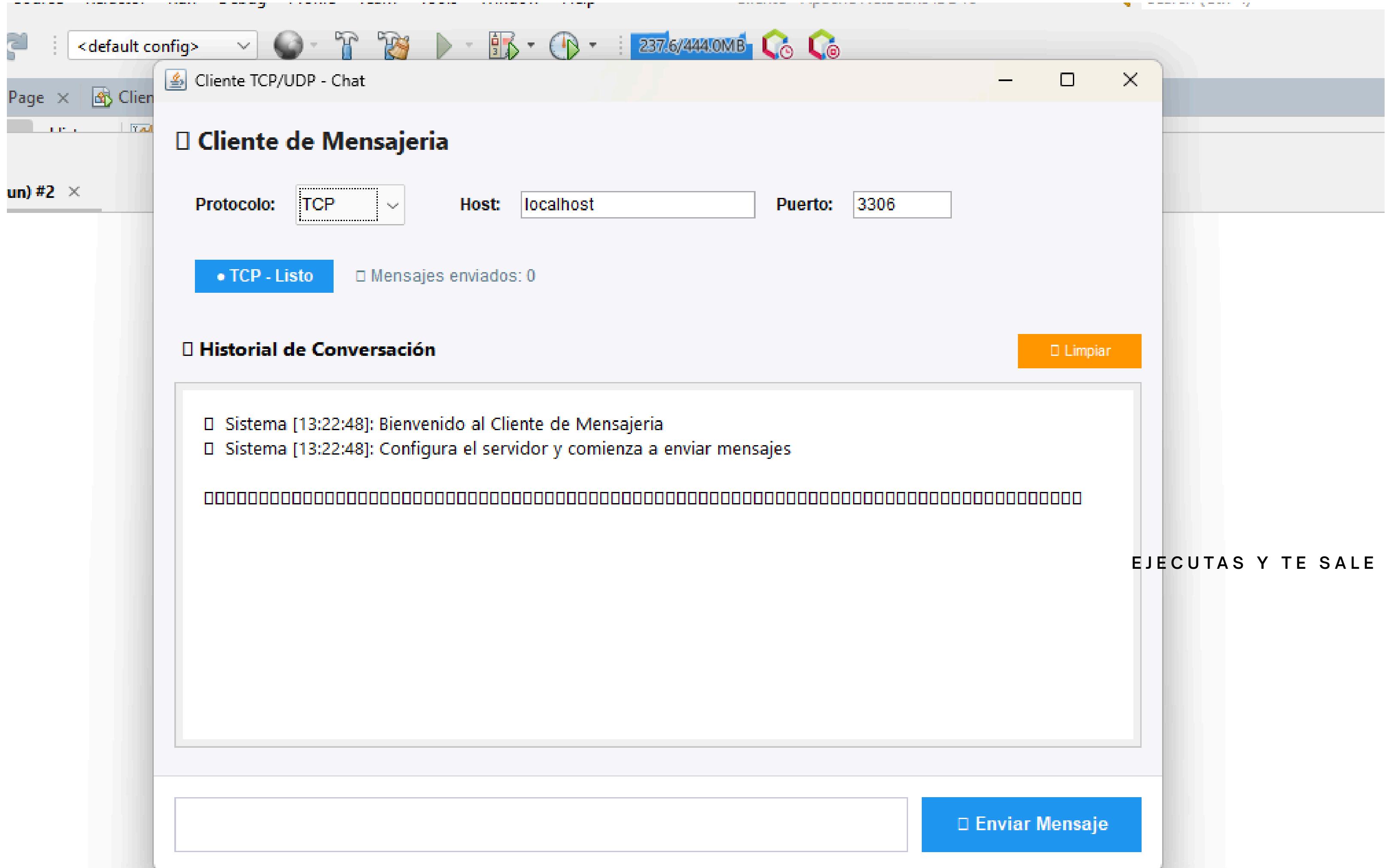
        // ENVIAR RESPUESTA
        PRINTWRITER SALIDA = NEW
        PRINTWRITER(SOCKET.GETOUTPUTSTREAM(), TRUE);
        SALIDA.PRINTLN("MENSAJE RECIBIDO: " +
MENSAJE);

        SOCKET.CLOSE();
        SYSTEM.OUT.PRINTLN("CONEXIÓN
CERRADA\n");
    }
}
```

SERVIDOR PRINCIPAL CODIGO PRINCIP

T¿ CÓMO FUNCIONA?

1. ABRE EL PUERTO 3306 PARA ESCUCHAR
2. ESPERA UN CLIENTE CON ACCEPT()
3. LEE EL MENSAJE DEL CLIENTE
4. ENVÍA UNA RESPUESTA AL CLIENTE
5. CIERRA LA CONEXIÓN
6. REPITE PARA EL SIGUIENTE CLIENTE



□ Cliente de Mensajería

Protocolo: TCP ▾

Host: localhost

Puerto: **3306**

• TCP - Listo □ Mensajes enviados: 1

Historial de Conversación

Limpiar

- Sistema [13:37:13]: Bienvenido al Cliente de Mensajería
- Sistema [13:37:13]: Configura el servidor y comienza a enviar mensajes

□ TÚ [13:37:35]
HOLA

□ Servidor [13:37:36]
Mensaje recibido: HOLA

**DESPUES DE EJECUTAR
LOS CODIGOS, PUEDES
MANDAR CUALQUIER
MENSAJE Y YA EL
SERVIDOR TE RESPONDE**

I U D Y