

## "Proyecto"

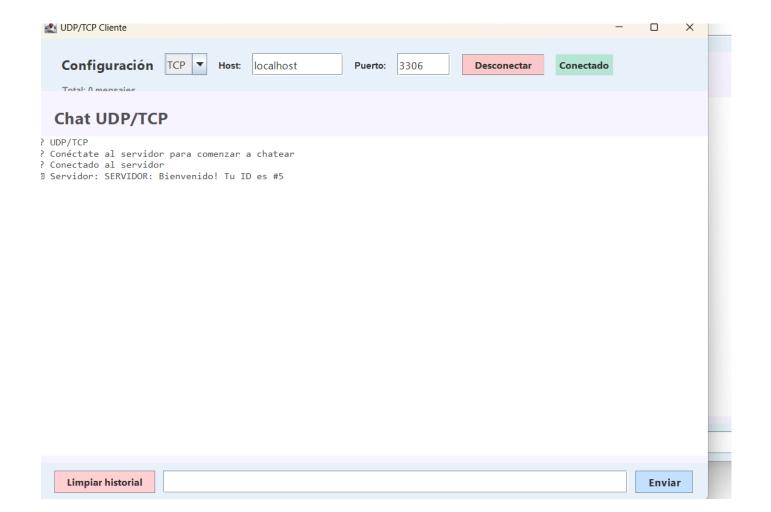
## MONTSERRAT OLAN LOPEZ

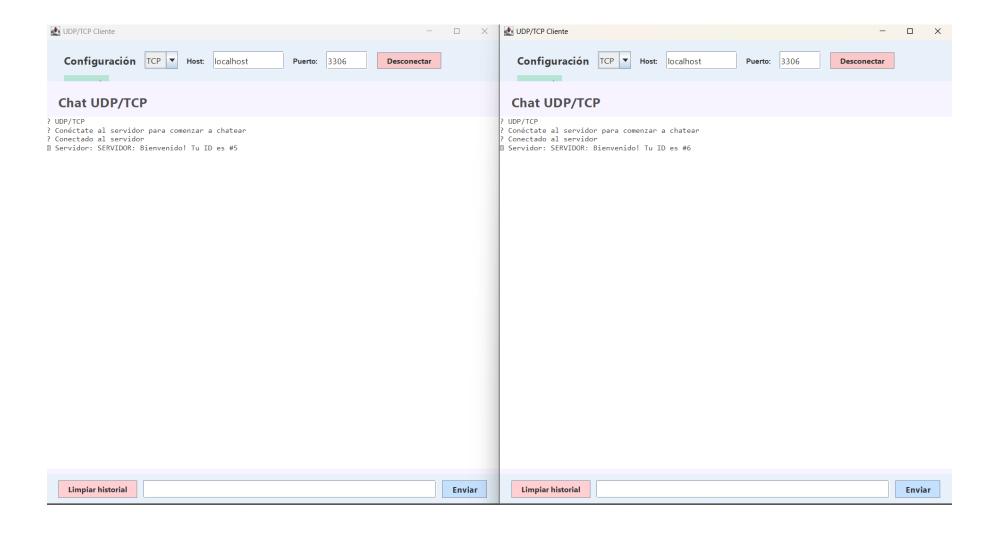
MODELO ENTIDAD-RELACIÓN, INSTITUTO UNIVERSITARIO DE YUCATÁN

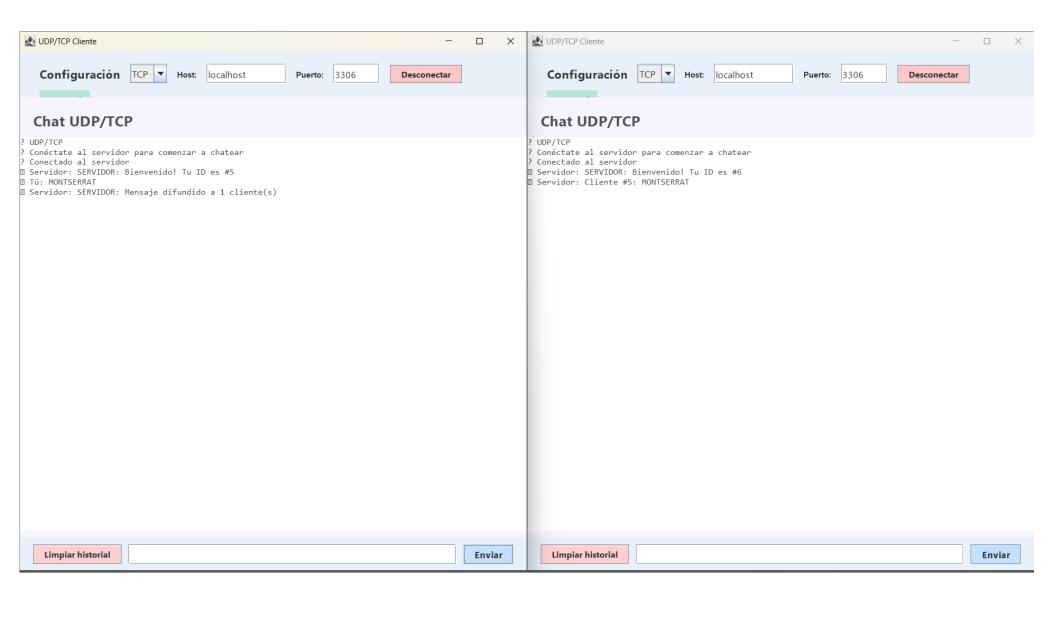
2303040768: REDES II

MAESTRA ING. PERLA ALEJANDRA LANDERO HEREDIA

19 DE OCTUBRE DEL 2025







```
🚳 Cliente.java 🗴 🌃 Servidor.java 🗴 🚳 Cliente2.java 🗴 🚳 Font.java 🗴 🚳 ComercioElectronico.java 🗴
       Source
  1
       package cliente;
  2
  3
       import java.io.*;
       import java.net.*;
  4
  5
       import javax.swing.*;
  6
       import java.awt.*;
       import java.awt.event.*;
  8
  9
       public class Cliente extends JFrame {
 10
           private JTextArea areaChat;
 11
           private JTextField campoMensaje, campoHost, campoPuerto;
 12
           private JButton btnEnviar, btnLimpiar, btnConectar;
 13
           private JComboBox<String> comboProtocolo;
 14
           private JLabel lblEstado, lblMensajesEnviados;
 15
           private JPanel panelEstadoConexion;
 16
 17
           private DatagramSocket socketUDP;
 18
           private Socket socketTCP;
 19
           private boolean conectado = false;
 20
           private boolean escuchando = false;
 21
           private String protocoloActual = "TCP";
           private String hostActual = "localhost";
 23
 24
           private int puertoActual = 3306;
           private int mensajesEnviados = 0;
 25
 26
 27
           public Cliente() {
               configurarVentana();
 28
               crearComponentes();
 29
 30
               configurarEventos();
 31
 32
           private void configurarVentana() {
 33
 34
               setTitle("UDP/TCP Cliente");
               setSize(900, 650);
 35
               setDefaultCloseOperation(JFrame. EXIT ON CLOSE);
 36
 37
               setLocationRelativeTo(null);
```

```
Cliente.java × Servidor.java × Cliente2.java × Font.java × ComercioElectronico.java ×
       Source
       package cliente;
    import java.io.*;
  2
       import java.net.*;
  3
       import java.util.*;
       import java.util.concurrent.*;
  6
  8
        * @author monts
 10
 11
       public class Servidor {
 12
 13
 2
           private static List<ManejadorClienteTCP> clientesTCP = new CopyOnWriteArrayList<>();
 Q<sub>A</sub>
           private static Map<String, ClienteUDP> clienteSUDP = new ConcurrentHashMap<>();
           private static int contadorClientes = 0;
 16
 17
           public static void main(String[] args) {
 18
 19
 20
 21
               servidorTCP();
 23
 24
           public static void servidorTCP() {
               final int PUERTO = 3306;
 26
 27
               try {
 28
                   ServerSocket servidor = new ServerSocket(PUERTO);
 29
                   System.out.println("
 30
                   System.out.println(" | Servidor TCP Multi-Cliente Iniciado
 31
                                                                                   ");
                   System.out.println(" | Puerto: " + PUERTO + "
                                                                                               ");
 32
                   System.out.println("-
                                                                           - \n");
 33
 34
 35
                   while (true) {
                       Socket socket = servidor.accept();
 36
                        contadorClientes++;
 37
```

```
Cliente.java × Servidor.java × Cliente2.java × Font.java × ComercioElectronico.java ×
package cliente;
   import java.io.*;
     import java.net.*;
  5
   - /**
  6
       * @author monts
      */
 8
 9
      public class Cliente2 {
 10
         public static void main(String[] args) {
 11 -
 12
 13
 14
 15
 16 -
          public static void clienteTCP() {
             final String HOST = "localhost";
 17
             final int PUERTO = 3306;
 18
 19
 20 =
             try {
 M
                 Socket socket = new Socket (HOST, PUERTO);
                 System.out.println("=======");
 22
                 System.out.println("CLIENTE 2 TCP CONECTADO");
 23
 24
                 System.out.println("========\n");
 25
                 BufferedReader teclado = new BufferedReader(
 26
 27
                    new InputStreamReader(System.in)
                 );
 28
 29
                 PrintWriter salida = new PrintWriter(socket.getOutputStream(), true);
 30
                 BufferedReader entrada = new BufferedReader(
 31
                    new InputStreamReader(socket.getInputStream())
 32
                 );
 33
 34
 35
 36
                 System.out.println("Escribe 'salir' para terminar\n");
 37
```