

Prueba de conexión de TCP:

CLIENTE JAVA :

```
package connect;
import java.io.*;
import java.net.*;
import java.net.*;
import java.util.*;
public class Cliente {
    private static String HOST = "localhost";
    private static int PUERTO = 8050;

    public static void main(String args[]) {

        InetAddress iaRemoteAddress = null;
        Socket socket;
        DataOutputStream mensaje;

        try {
            //Creamos nuestro socket
            socket = new Socket(HOST, PUERTO);

            mensaje = new DataOutputStream(socket.getOutputStream());

            //Enviamos un mensaje

            mensaje.writeUTF("Hola!!<EOF>");

            //Cerramos la conexión
            socket.close();

        } catch (UnknownHostException e) {
            System.out.println("El host no existe o no está activo.");
        } catch (IOException e) {
            System.out.println("Error de entrada/salida."+ e);
        }

    }
}
```

SERVIDOR EN PYTHON

```
#!/usr/bin/env python
#Se importa el módulo
import socket
```

```

#instanciamos un objeto para trabajar con el socket
ser = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

#Puerto y servidor que debe escuchar
ser.bind(("", 8050))

#Aceptamos conexiones entrantes con el metodo listen. Por parámetro las
conexiones simultáneas.
ser.listen(1)

#Instanciamos un objeto cli (socket cliente) para recibir datos
cli, addr = ser.accept()

while True:

    #Recibimos el mensaje, con el metodo recv recibimos datos. Por parametro
la cantidad de bytes para recibir
    recibido = cli.recv(1024)
    print(addr)
    #Si se reciben datos nos muestra la IP y el mensaje recibido
    print ("Recibo conexion de la IP: " + str(addr[0]) + " Puerto: " +
str(addr[1]))
    print ("Recibo conexion de la IP: " + recibido.decode())
    #Devolvemos el mensaje al cliente
    mensaje="recibido"
    cli.send(mensaje.encode())

#Cerramos la instancia del socket cliente y servidor
cli.close()
ser.close()

print("Conexiones cerradas")

}

```

PRUEBA:

```

rs\2daw3\.vscode\extensions\ms-python.python-2022.14.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '569
97' '--' 'c:\Users\2daw3\Desktop\cliente-servidor\Servidor-tcp.py'
('127.0.0.1', 56932)
Recibo conexion de la IP: 127.0.0.1 Puerto: 56932
Recibo conexion de la IP: ¡Hola!!<EOF>
('127.0.0.1', 56932)
Recibo conexion de la IP: 127.0.0.1 Puerto: 56932
Recibo conexion de la IP:

```

## Conexión UDP

### Cliente UDP

```
package UPDdisc;
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.net.UnknownHostException;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Cliente {

    public static void main(String[] args) {

        //puerto del servidor
        final int PUERTO_SERVIDOR = 5000;
        //buffer donde se almacenara los mensajes
        byte[] buffer = new byte[1024];

        try {
            //Obtengo la localizacion de localhost
            InetAddress direccionServidor =
InetAddress.getByName("localhost");

            //Creo el socket de UDP
            DatagramSocket socketUDP = new DatagramSocket();

            String mensaje = "¡Hola mundo desde el cliente!";

            //Convierto el mensaje a bytes
            buffer = mensaje.getBytes();

            //Creo un datagrama
            DatagramPacket pregunta = new DatagramPacket(buffer,
buffer.length, direccionServidor, PUERTO_SERVIDOR);

            //Lo envio con send
            System.out.println("Envio el datagrama");
            socketUDP.send(pregunta);

            //Preparo la respuesta
            DatagramPacket peticion = new DatagramPacket(buffer,
buffer.length);

            //Recibo la respuesta
```

```
        socketUDP.receive(peticion);
        System.out.println("Recibo la peticion");

        //Cojo los datos y lo muestro
        mensaje = new String(peticion.getData());
        System.out.println(mensaje);

        //cierro el socket
        socketUDP.close();

    } catch (SocketException ex) {
        Logger.getLogger(Cliente.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (UnknownHostException ex) {
        Logger.getLogger(Cliente.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (IOException ex) {
        Logger.getLogger(Cliente.class.getName()).log(Level.SEVERE, null,
ex);
    }
}
}
```

## Servidor UDP

```
package UPDdisc;
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Servidor {

    public static void main(String[] args) {

        final int PUERTO = 5000;
        byte[] buffer = new byte[1024];

        try {
            System.out.println("Iniciado el servidor UDP");
            //Creacion del socket
            DatagramSocket socketUDP = new DatagramSocket(PUERTO);

            //Siempre atendera peticiones
            while (true) {

                //Preparo la respuesta
                DatagramPacket peticion = new DatagramPacket(buffer,
buffer.length);

                //Recibo el datagrama
                socketUDP.receive(peticion);
                System.out.println("Recibo la informacion del cliente");

                //Convierto lo recibido y mostrar el mensaje
                String mensaje = new String(peticion.getData());
                System.out.println(mensaje);

                //Obtengo el puerto y la direccion de origen
                //Sino se quiere responder, no es necesario
                int puertoCliente = peticion.getPort();
                InetAddress direccion = peticion.getAddress();

                mensaje = "¡Hola mundo desde el servidor!";
                buffer = mensaje.getBytes();

                //creo el datagrama
                DatagramPacket respuesta = new DatagramPacket(buffer,
buffer.length, direccion, puertoCliente);
```

```

        //Envio la información
        System.out.println("Envio la informacion del cliente");
        socketUDP.send(respuesta);

    }

    } catch (SocketException ex) {
        Logger.getLogger(Servidor.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (IOException ex) {
        Logger.getLogger(Servidor.class.getName()).log(Level.SEVERE, null,
ex);
    }

    }

}
}

```

Prueba:

