

Estructura básica conexión

Servidor

```
package data;

import java.io.*;
import java.net.*;
import java.util.Scanner;

public class Servidor {

    public static void main(String[] args) {
        ServerSocket servidor;
        DataInputStream in;
        DataOutputStream out;
        try {
            //Creacion de servidor
            servidor = new ServerSocket(5000);
            System.out.println("Servidor Abierto, esperando...");
            Socket cliente = servidor.accept();//Esperando conexión
            System.out.println("Cliente conectado");
            System.out.println("Esperando mensaje del cliente...");

            in = new
DataInputStream(cliente.getInputStream());//Recibir mensajes
            out = new
DataOutputStream(cliente.getOutputStream());//Enviar mensajes

            String mensaje = in.readUTF();//Leemos el mensaje estamos
a la espera
            System.out.println("Mensaje recibido del cliente:
"+mensaje);

            System.out.print("Mensaje que enviaremos al cliente: ");
            out.writeUTF(new Scanner(System.in).nextLine());

            cliente.close();//Cerramos cliente
            System.out.println("Cliente desconectado");

        } catch (IOException e) {
            System.out.println(e);
        }
    }
}
```

Cliente:

```
package data;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.util.Scanner;

public class Cliente {

    public static void main(String[] args) {
        Socket cliente;
        DataInputStream in;
        DataOutputStream out;
        try {

            cliente = new Socket("localhost", 5000);
            in = new
DataInputStream(cliente.getInputStream()); //Recibir mensajes
            out = new
DataOutputStream(cliente.getOutputStream()); //Enviar mensajes

            System.out.print("Mensaje que enviaremos al servidor: ");
            out.writeUTF(new Scanner(System.in).nextLine());

            System.out.println("Esperando respuesta del servidor...");
            String mensaje=in.readUTF();
            System.out.println("Mensaje recibido del servidor:
"+mensaje);

            //Cerramos el cliente
            cliente.close();

        } catch (IOException io) {
            System.out.println(io);
        }
    }
}
```

Orden:

El cliente envía un mensaje, luego el servidor lo lee y responde, por ultimo el cliente lo recibe.

Salida:

```
Output - Red 4 (run) x run:
run: Mensjae que enviaremos al servidor:
Servidor Abierto, esperando...
```

```
run:
Servidor Abierto, esperando...
Cliente conectado
Esperando mensaje del cliente...
```

```
run:
Mensjae que enviaremos al servidor: Soy Kevin desde cliente
Esperando respuesta del servidor...
|
```

```
run:
Servidor Abierto, esperando...
Cliente conectado
Esperando mensaje del cliente...
Mensaje recibido del cliente: Soy Kevin desde cliente
Mensaje que enviaremos al cliente:
```

```
run:
Servidor Abierto, esperando...
Cliente conectado
Esperando mensaje del cliente...
Mensaje recibido del cliente: Soy Kevin desde cliente
Mensaje que enviaremos al cliente: Kevin desde el server
Cliente desconectado
BUILD SUCCESSFUL (total time: 2 minutes 46 seconds)
|
```

```
run:
Mensjae que enviaremos al servidor: Soy Kevin desde cliente
Esperando respuesta del servidor...
Mensaje recibido del servidor: Kevin desde el server
BUILD SUCCESSFUL (total time: 2 minutes 24 seconds)
|
```