

Resultados dos Testes

Cliente UDP → Servidor UDP

```
~/Dropbox/socket (0.344s)
javac UDPCliient.java && java UDPCliient "0ie" 127.0.0.1
Reply: 0ie
```

```
Last login: Sat Mar  2 22:29:22 on console
alisson@Alissons-MacBook-Air socket % javac UDPServer.java && java UDPServer
```

```

1349 39.849645    127.0.0.1      127.0.0.1      UDP          35 59812 → 6789 Len=3
1350 39.849951    127.0.0.1      127.0.0.1      UDP          35 6789 → 59812 Len=3
1351 48.286828     127.0.0.1      127.0.0.1      TCP          68 4978 → 62485 [SYN] Seq=0 Win=65535 Len=0 MSS=16344 WS=64 TSval=1532261505 TSecr= SACK_PERM
1352 48.286187     127.0.0.1      127.0.0.1      TCP          68 62485 → 49781 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=16344 WS=64 TSval=1746369496 TSecr=1532261505 SACK_PERM
1353 48.286208     127.0.0.1      127.0.0.1      TCP          56 49781 → 62485 [ACK] Seq=1 Ack=1 Win=480256 Len=0 TSval=1532261505 TSecr=1746369496
1354 48.286226     127.0.0.1      127.0.0.1      TCP          56 [TCP Window Update] 62485 → 49781 [ACK] Seq=1 Ack=1 Win=480256 Len=0 TSval=1746369496 TSecr=1532261505

> Frame 1349: 35 bytes on wire (280 bits), 35 bytes captured (280 bits) on interface lo0, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> User Datagram Protocol, Src Port: 59812, Dst Port: 6789
> Data (3 bytes)
0000 02 00 00 00 45 00 00 1f 34 5d 00 00 04 11 00 00 ...E... 4)...@...
0010 7f 00 00 01 7f 00 00 01 e9 a4 1a 85 00 0b fe 1e ........
0020 4f 69 65                                     Oie

1349 39.849645    127.0.0.1      127.0.0.1      UDP          35 59812 → 6789 Len=3
1350 39.849951    127.0.0.1      127.0.0.1      UDP          35 6789 → 59812 Len=3
1351 48.286828     127.0.0.1      127.0.0.1      TCP          68 49781 → 62485 [SYN] Seq=0 Win=65535 Len=0 MSS=16344 WS=64 TSval=1532261505 TSecr= SACK_PERM
1352 48.286187     127.0.0.1      127.0.0.1      TCP          68 62485 → 49781 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=16344 WS=64 TSval=1746369496 TSecr=1532261505 SACK_PERM
1353 48.286208     127.0.0.1      127.0.0.1      TCP          56 49781 → 62485 [ACK] Seq=1 Ack=1 Win=480256 Len=0 TSval=1532261505 TSecr=1746369496
1354 48.286226     127.0.0.1      127.0.0.1      TCP          56 [TCP Window Update] 62485 → 49781 [ACK] Seq=1 Ack=1 Win=480256 Len=0 TSval=1746369496 TSecr=1532261505

> Frame 1350: 35 bytes on wire (280 bits), 35 bytes captured (280 bits) on interface lo0, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> User Datagram Protocol, Src Port: 6789, Dst Port: 59812
> Data (3 bytes)
0000 02 00 00 00 45 00 00 1f 6b e4 00 00 04 11 00 00 ...E... k)...@...
0010 7f 00 00 01 7f 00 00 01 1a 85 e9 a4 00 0b fe 1e ........
0020 4f 69 65                                     Oie

```

Cliente UDP → Servidor TCP

```
~/Dropbox/socket  
javac UDPCClient.java && java UDPCClient "0ie" 127.0.0.1
```

```
alisson@Alissons-MacBook-Air socket % javac TCPServer.java && java TCPServer
```

```

9 26.3181864 127.0.0.1 127.0.0.1 ICMP 60 Destination unreachable (Port unreachable)
10 26.312500 127.0.0.1 127.0.0.1 UDP 221 51796 - 32376 Len=189
11 26.312533 127.0.0.1 127.0.0.1 ICMP 60 Destination unreachable (Port unreachable)
12 34.664130 fe80::103:f0ff:fe_ f02::fb MMS 129 Standard query 0x0000 PTR_companion-link_tcp.local, "QM" question PTR CLink-731132b2b19e_companion-link_tcp.local

> Frame 11: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface lo0, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Internet Control Message Protocol

```

Cliente TCP → Servidor TCP

```
~/Dropbox/socket (0.472s)
yes | javac TCPCClient.java && java TCPCClient "0ie" 127.0.0.1
Received: 0ie
```

```
alisson@Alissons-MacBook-Air socket % javac TCPServer.java && java TCPServer
```

1	0.000000	127.0.0.1	127.0.0.1	TCP	68	52456 → 7896 [SYN] Seq=0 Win=65535 Len=0 MSS=16344 WS=64 TSval=3879983456 TSecr=0 SACK_PERM	
2	0.000159	127.0.0.1	127.0.0.1	TCP	68	7896 → 52456 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=16344 WS=64 TSval=1255305102 TSecr=3879983456 SACK_PERM	
3	0.000182	127.0.0.1	127.0.0.1	TCP	56	52456 → 7896 [ACK] Seq=1 Ack=1 Win=408256 Len=0 TSval=3879983456 TSecr=1255305102	
4	0.000196	127.0.0.1	127.0.0.1	TCP	56	[TCP Window Update] 7896 → 52456 [ACK] Seq=1 Ack=1 Win=408256 Len=0 TSval=1255305102 TSecr=3879983456	
5	0.001327	127.0.0.1	127.0.0.1	TCP	61	52456 → 7896 [PSH, ACK] Seq=1 Ack=1 Win=408256 Len=5 TSval=3879983458 TSecr=1255305102	
6	0.001355	127.0.0.1	127.0.0.1	TCP	56	7896 → 52456 [ACK] Seq=1 Ack=6 Win=408256 Len=0 TSval=1255305104 TSecr=3879983458	
7	0.003154	127.0.0.1	127.0.0.1	TCP	61	7896 → 52456 [PSH, ACK] Seq=1 Ack=6 Win=408256 Len=5 TSval=1255305105 TSecr=3879983458	
8	0.003184	127.0.0.1	127.0.0.1	TCP	56	52456 → 7896 [ACK] Seq=6 Ack=6 Win=408256 Len=0 TSval=3879983459 TSecr=1255305105	
9	0.003269	127.0.0.1	127.0.0.1	TCP	56	7896 → 52456 [FIN, ACK] Seq=6 Ack=6 Win=408256 Len=0 TSval=1255305105 TSecr=3879983459	
10	0.003296	127.0.0.1	127.0.0.1	TCP	56	52456 → 7896 [ACK] Seq=6 Ack=7 Win=408256 Len=0 TSval=3879983459 TSecr=1255305105	
11	0.004641	127.0.0.1	127.0.0.1	TCP	56	52456 → 7896 [FIN, ACK] Seq=6 Ack=7 Win=408256 Len=0 TSval=3879983461 TSecr=1255305105	
12	0.004686	127.0.0.1	127.0.0.1	TCP	56	7896 → 52456 [ACK] Seq=7 Ack=7 Win=408256 Len=0 TSval=1255305107 TSecr=3879983461	

> Frame 1: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface lo0, id 0

> Null/Loopback

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> Transmission Control Protocol, Src Port: 52456, Dst Port: 7896, Seq: 0, Len: 0

0000 02 00 00 00 45 00 00 40 00 00 40 00 40 06 00 00E..@..@...

0010 7f 00 00 01 7f 00 00 01 cc e8 1e d8 f6 eb ef a7>.....

0020 00 00 00 00 b0 02 ff ff fe 34 00 00 02 04 3f d84....?..

0030 01 03 03 06 01 01 08 0a e7 43 d9 60 00 00 00 00C.....

0040 04 02 00 00

Cliente TCP ➔ Servidor UDP

```
nvim TCPClient.java
```

```
~/Dropbox/socket (0.345s)
```

```
javac TCPClient.java && java TCPClient "0ie" 127.0.0.1
IO:Connection refused
```

```
Last login: Sat Mar 2 22:29:22 on console
```

```
alisson@Alissons-MacBook-Air socket % javac UDPServer.java && java UDPServer
```

162	4.514054	127.0.0.1	127.0.0.1	TCP	56	62485 → 51102 [ACK] Seq=361 Ack=238 Win=408064 Len=0 TSval=1265972824 TSecr=1236456586	
163	4.514149	127.0.0.1	127.0.0.1	TCP	56	62485 → 51102 [FIN, ACK] Seq=361 Ack=238 Win=408064 Len=0 TSval=1265972824 TSecr=1236456586	
164	4.514176	127.0.0.1	127.0.0.1	TCP	56	51102 → 62485 [ACK] Seq=238 Ack=362 Win=407936 Len=0 TSval=1236456586 TSecr=1265972824	
165	4.804636	127.0.0.1	127.0.0.1	TCP	68	51103 → 7896 [SYN] Seq=0 Win=65535 Len=0 MSS=16344 WS=64 TSval=1870556523 TSecr=0 SACK_PERM	
166	4.804690	127.0.0.1	127.0.0.1	TCP	44	7896 → 51103 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0	
167	5.014315	127.0.0.1	127.0.0.1	TCP	68	51104 → 62485 [SYN] Seq=0 Win=65535 Len=0 MSS=16344 WS=64 TSval=947106567 TSecr=0 SACK_PERM	
168	5.014430	127.0.0.1	127.0.0.1	TCP	68	62485 → 51104 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=16344 WS=64 TSval=2971197031 TSecr=947106567 SACK_PERM	

> Frame 165: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface lo0, id 0

> Null/Loopback

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> Transmission Control Protocol, Src Port: 51103, Dst Port: 7896, Seq: 0, Len: 0

0000 02 00 00 00 45 00 00 40 00 00 40 00 40 06 00 00E..@..@...

0010 7f 00 00 01 7f 00 00 01 c7 9f 1e d8 3e 29 d8 05>.....

0020 00 00 00 00 b0 02 ff ff fe 34 00 00 02 04 3f d84....?..

0030 01 03 03 06 01 01 08 0a 6f 7e 6d 6b 00 00 00 00o-mk....

0040 04 02 00 00

Codigo Usado

```
import java.net.*;
import java.io.*;

public class UDPClient {
    public static void main(String args[]) {
        // args fornece o conteúdo da mensagem e o nome de host do servidor
        DatagramSocket aSocket = null;
        try {
            aSocket = new DatagramSocket();
            byte[] m = args[0].getBytes();
            InetAddress aHost = InetAddress.getByName(args[1]);
            int serverPort = 6789;
            DatagramPacket request = new DatagramPacket(m, m.length, aHost,
serverPort); aSocket.send(request);
            byte[] buffer = new byte[1000];
            DatagramPacket reply = new DatagramPacket(buffer, buffer.length);
            aSocket.receive(reply);
            System.out.println("Reply: " + new String(reply.getData()));
        } catch (SocketException e) {
            System.out.println("Socket: " + e.getMessage());
        } catch (IOException e) {
            System.out.println("IO: " + e.getMessage());
        } finally {
            if (aSocket != null)
                aSocket.close();
        }
    }
}
```

```

import java.net.*;
import java.io.*;

public class TCPClient {
    public static void main(String args[]) {
        // Os argumentos fornecem a mensagem e o nome de host de destino
        Socket s = null;
        try {
            int serverPort = 7896;
            s = new Socket(args[1], serverPort); // Conexão com o servidor especificado
            DataInputStream in = new DataInputStream(s.getInputStream());
            DataOutputStream out = new DataOutputStream(s.getOutputStream());

            out.writeUTF(args[0]); // Envia a mensagem para o servidor

            String data = in.readUTF(); // Aguarda a resposta do servidor
            System.out.println("Received: " + data); // Exibe a mensagem recebida do servidor

        } catch (UnknownHostException e) {
            System.out.println("Sock: " + e.getMessage()); // Trata exceção de host desconhecido
        } catch (EOFException e) {
            System.out.println("EOF: " + e.getMessage()); // Trata exceção de fim de arquivo
        } catch (IOException e) {
            System.out.println("IO: " + e.getMessage()); // Trata exceção de entrada/saída
        } finally {
            // Fecha a conexão com o servidor
            if (s != null)
                try {
                    s.close();
                } catch (IOException e) {
                    /* Falha ao fechar a conexão */
                }
        }
    }
}

```

```
import java.net.*;
import java.io.*;

public class UDPServer {
    public static void main(String args[]) {
        DatagramSocket aSocket = null;
        try {
            aSocket = new DatagramSocket(6789);
            byte[] buffer = new byte[1000];
            while (true) {
                DatagramPacket request = new DatagramPacket(buffer, buffer.length);
                aSocket.receive(request);
                // Verificar se dados foram recebidos
                if (request.getLength() > 0) {
                    // Criar pacote pra responder dados recebidos
                    DatagramPacket reply = new DatagramPacket(request.getData(),
                        request.getLength(), request.getAddress(),
                        request.getPort()); aSocket.send(reply);
                } else {
                    // Mostrar erro
                    System.out.println("No data received from client.");
                }
            }
        } catch (SocketException e) {
            System.out.println("Socket: " + e.getMessage());
        } catch (IOException e) {
            System.out.println("IO: " + e.getMessage());
        } finally {
            if (aSocket != null) aSocket.close();
        }
    }
}
```

```

import java.net.*;
import java.io.*;

public class TCPServer {
    public static void main(String args[]) {
        try {
            int serverPort = 7896;
            // Cria um socket de servidor e o associa à porta especificada
            ServerSocket listenSocket = new ServerSocket(serverPort);

            // Loop infinito para aceitar conexões de clientes
            while (true) {
                // Aguarda e aceita a conexão de um cliente
                Socket clientSocket = listenSocket.accept();

                // Inicia uma nova thread para lidar com a conexão do cliente
                Connection c = new Connection(clientSocket);
            }
        } catch (IOException e) {
            System.out.println("Listen : " + e.getMessage());
        }
    }
}

class Connection extends Thread {
    DataInputStream in;
    DataOutputStream out;
    Socket clientSocket;

    public Connection(Socket aClientSocket) {
        try {
            clientSocket = aClientSocket;
            // Cria fluxos de entrada e saída para comunicação com o cliente
            in = new DataInputStream(clientSocket.getInputStream());
            out = new DataOutputStream(clientSocket.getOutputStream());
            this.start();
        } catch (IOException e) {
            System.out.println("Connection: " + e.getMessage());
        }
    }

    public void run() {
        try {
            // Lê os dados enviados pelo cliente e os envia de volta
            String data = in.readUTF();
            out.writeUTF(data);
        } catch (EOFException e) {
            System.out.println("EOF: " + e.getMessage());
        } catch (IOException e) {
            System.out.println("IO: " + e.getMessage());
        } finally {
            try {
                // Fecha o socket do cliente após a comunicação
                clientSocket.close();
            } catch (IOException e) {
                // TODO: tratamento de erro se o fechamento do socket falhar
            }
        }
    }
}

```