

**Aluno: Rainan Gramacho**

**Matrícula: 164161020**

**////////////////UDP CLIENT.java////////////////**

```
import java.net.DatagramPacket;
```

```
import java.net.DatagramSocket;
```

```
import java.net.InetAddress;
```

```
import java.util.Scanner;
```

```
public class UDP_Client{
```

```
    public static void main(String args[]) throws Exception{
```

```
        // c:\> java UDPClient1 <mensagem> <endDst>
```

```
        DatagramSocket dsocket = new DatagramSocket();
```

```
        Scanner input = new Scanner(System.in);
```

```
        String mensagemSair = "Cliente Saindo...";
```

```
        byte[] msgVai = new byte [1024];
```

```
        byte[] endD = new byte [1024];
```

```
        byte[] portArg = new byte [1024];
```

```
        String mensagemVai = null;
```

```
        String endDestino = null;
```

```
        InetAddress endDst = null;
```

```
        String endString = null;
```

```
        String portArgInt = null;
```

```
if(args.length == 0){ // se nao tiver arg
```

```
    mensagemVai = "ping";
```

```
    msgVai = mensagemVai.getBytes();
```

```
} else if(args.length >=1){ // se tiver um arg
```

```
    msgVai = args[0].getBytes();
```

```
}
```

```
if(args.length <=1){ // se nao tiver arg ou se tiver so o primeiro
```

arg

```
    endDestino = "localhost";
```

```
    endDst = InetAddress.getByName(endDestino);
```

```
} else if(args.length >1){ // se tiver o primeiro arg e o segundo
```

arg

```
    endD = args[1].getBytes();
```

```
    endString = new String(endD);
```

```
    endDst = InetAddress.getByName(endString);
```

```
}
```

```
int portaDst = 0;

if(args.length <=2){// se nao tiver arg, ou se tiver so o
primeiro arg,ou se tiver o primeiro e o segundo arg

    portaDst = 6789;

}

}

else if(args.length >2){
    portArg = args[2].getBytes();
    portArgInt = new String(portArg);
    portaDst = Integer.parseInt(portArgInt);
}

}

try{
    //byte[] msgVai = args[0].getBytes();
    //byte[] msgVai = mensagemVai.getBytes(); // tirando a opção
de passar mensagem por argumento, agr sera passada pelo scanner

    //InetAddress endDst = InetAddress.getByName(endDestino);
    //int portaDst = 6789;

    DatagramPacket pctVai = new DatagramPacket(msgVai,
msgVai.length, endDst, portaDst);
    dsocket.send(pctVai);
}
```

```

        System.out.println("Enviei solicitacao ao Servidor\n");

        byte[] msgVem = new byte[1024];

        DatagramPacket pctVem = new DatagramPacket(msgVem,
msgVem.length);

        dsocket.receive(pctVem);

        System.out.println("Recebi resposta do Servidor:\n");

        System.out.println("Chegou: " + new
String(pctVem.getData()));

        Thread.sleep(5000);

        dsocket.close();

    }    catch(Exception e){

    }

}

}

```

**////////////////////////////////UDP\_Server.java //////////////////////////////////**

```

import java.io.IOException;

import java.net.SocketException;

```

```
import java.util.InputMismatchException;
```

```
import java.net.DatagramPacket;
```

```
import java.net.DatagramSocket;
```

```
import java.util.Scanner;
```

```
import java.util.ArrayList;
```

```
public class UDP_Server{
```

```
    public static void main(String args[]) throws Exception {
```

```
        //DatagramSocket ds = new DatagramSocket(6789);
```

```
        DatagramSocket dsocket = null;
```

```
        Scanner input = new Scanner(System.in);
```

```
        int porta =0;
```

```
        String portServerString = "";
```

```
        byte[] portServerArg = new byte [1024];
```

```
        if(args.length == 0){
```

```
            porta = 6789;
```

```
        } else if(args.length >=1){
```

```
            portServerArg = args[0].getBytes();
```

```
            portServerString = new String(portServerArg);
```

```
            porta = Integer.parseInt(portServerString);
```

```
        }
```

```

try {

    dsocket = new DatagramSocket(porta);

    System.out.printf("Aguardando Cliente na porta
%d...\n",dsocket.getLocalPort());

    //int clientes = 0;
    byte[] msg = new byte[1024];

    ArrayList<UDPServerThread> threads = new ArrayList<>();

new Thread(new Runnable() {
    @Override
    public void run() {

        while (true) {
            int clientes = 0;
            try {
                for (UDPServerThread thread : threads) {
                    if (thread.isAlive()) {
                        clientes++;
                    }
                }
            }
        }
    }
}

```

```
        System.out.println("Conexões ativas: " + clientes);  
        Thread.sleep(2000);  
    } catch (InterruptedException e) {  
  
    }  
}  
}  
}).start();
```

```
        while(true){  
            DatagramPacket pctVeio = new DatagramPacket(msg,  
msg.length);  
            dsocket.receive(pctVeio);  
  
            System.out.println("Recebi solicitacao do Cliente:"  
+pctVeio.getAddress());  
  
            UDPServerThread thread = new UDPServerThread(pctVeio);  
            threads.add(thread);  
            thread.start();  
  
            System.out.println("Enviei resposta ao Cliente\n");  
            //Thread.sleep(2000);  
            //new UDPServerThread(pctVeio).start();  
  
            //clientes ++;
```

```

        //System.out.printf("Estou atendendo %d ",clientes);

    }
    }catch(Exception e){
    }
    //ds.close();

}

}

```

**////////////////UDPServerThread.java////////////////**

```

import java.net.DatagramPacket;
import java.net.DatagramSocket;

class UDPServerThread extends Thread{

    private DatagramPacket pctVeio;
    //private DatagramPacket pctVai;

```



```

    public UDPServerThread(DatagramPacket pctVeio){
        this.pctVeio = pctVeio;
    }

    @Override
    public void run(){
        // essa linha veio do UDPServer3

        DatagramPacket pctVai = new
DatagramPacket(pctVeio.getData(),pctVeio.getLength(),pctVeio
.getAddress(),pctVeio.getPort());

        try{
            DatagramSocket dst = new DatagramSocket();
            dst.send(pctVai); // essa linha veio do
UDPServer3

            Thread.sleep(6000);//esse sleep faz com que
dê tempo de mostrar a quantidade de clientes ativos

            dst.close();
        }catch(Exception e){

        }
    }
}

```