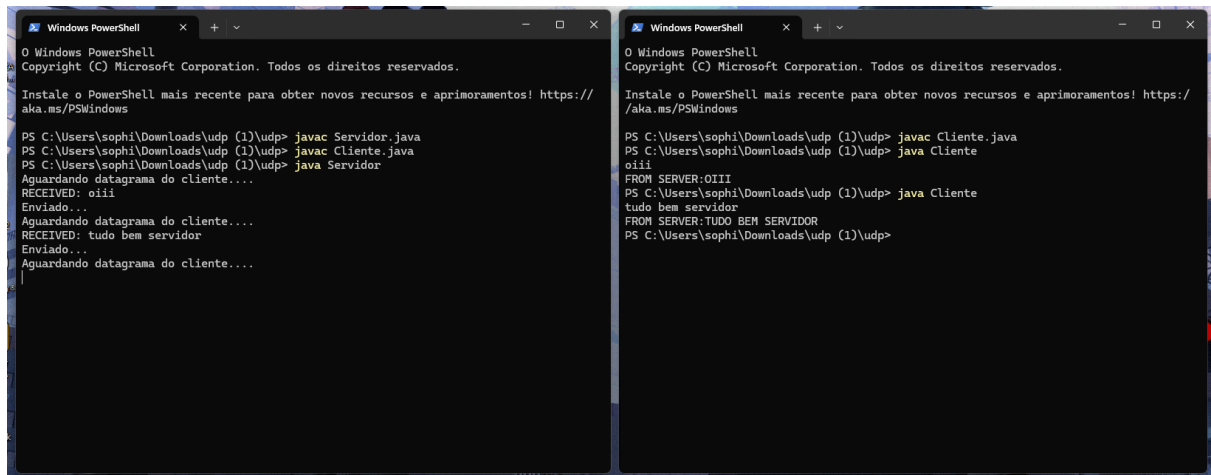


Lista 05 - Aplicação com UDP em Java

Sophia Carrazza - CC PUC Minas



```
O Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

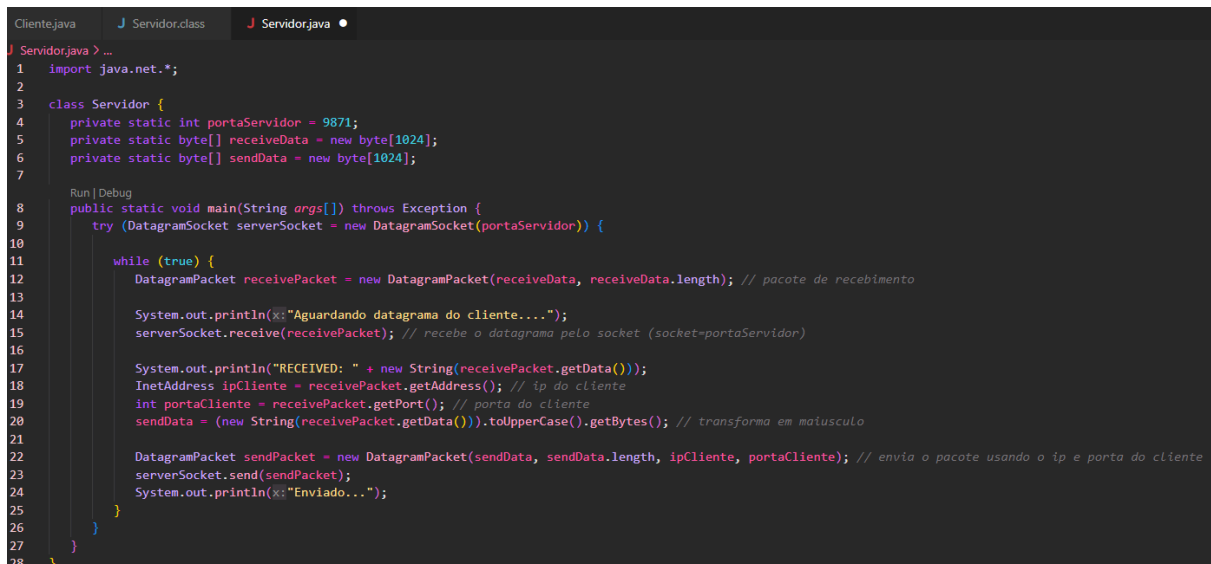
Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! https://aka.ms/PSWindows

PS C:\Users\sophi\Downloads\udp (1)\udp> javac Servidor.java
PS C:\Users\sophi\Downloads\udp (1)\udp> javac Cliente.java
PS C:\Users\sophi\Downloads\udp (1)\udp> java Servidor
Aguardando datagrama do cliente....
RECEIVED: oi
Enviado...
Aguardando datagrama do cliente....
RECEIVED: tudo bem servidor
Enviado...
Aguardando datagrama do cliente....

O Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! https://aka.ms/PSWindows

PS C:\Users\sophi\Downloads\udp (1)\udp> javac Cliente.java
PS C:\Users\sophi\Downloads\udp (1)\udp> java Cliente
oi
FROM SERVER:oi
PS C:\Users\sophi\Downloads\udp (1)\udp> java Cliente
tudo bem servidor
FROM SERVER:TUDO BEM SERVIDOR
PS C:\Users\sophi\Downloads\udp (1)\udp>
```



```
Cliente.java  Servidor.class  Servidor.java
Servidor.java >...
1  import java.net.*;
2
3  class Servidor {
4      private static int portaServidor = 9871;
5      private static byte[] receiveData = new byte[1024];
6      private static byte[] sendData = new byte[1024];
7
8      public static void main(String args[]) throws Exception {
9          try (DatagramSocket serverSocket = new DatagramSocket(portaServidor)) {
10
11              while (true) {
12                  DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length); // pacote de recebimento
13
14                  System.out.println("Aguardando datagrama do cliente....");
15                  serverSocket.receive(receivePacket); // recebe o datagrama pelo socket (socket=portaServidor)
16
17                  System.out.println("RECEIVED: " + new String(receivePacket.getData()));
18                  InetAddress ipCliente = receivePacket.getAddress(); // ip do cliente
19                  int portaCliente = receivePacket.getPort(); // porta do cliente
20                  sendData = (new String(receivePacket.getData()).toUpperCase()).getBytes(); // transforma em maiusculo
21
22                  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, ipCliente, portaCliente); // envia o pacote usando o ip e porta do cliente
23                  serverSocket.send(sendPacket);
24                  System.out.println("Enviado...");
25              }
26          }
27      }
28  }
```

```
Cliente.java x Servidor.class Servidor.java
Cliente.java > Cliente > lerString()
1 import java.io.*;
2 import java.net.*;
3
4 class Cliente {
5     private static int portaServidor = 9871;
6     private static byte[] sendData = new byte[1024];
7     private static byte[] receiveData = new byte[1024];
8
9     public static byte[] lerString() throws Exception {
10         BufferedReader in = new BufferedReader(new InputStreamReader(System.in)); // entrada de dados
11         return in.readLine().getBytes();
12     }
13
14     public static void main(String args[]) throws Exception {
15         DatagramSocket clientSocket = new DatagramSocket(); // cria o socket (porta aleatória)
16         InetAddress ipServidor = InetAddress.getByName(host:"192.168.0.117"); // ip do servidor
17         sendData = lerString(); // lê a string do teclado
18
19         DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, ipServidor, portaServidor); // cria o
20                                                                                                     // pacote
21         clientSocket.send(sendPacket); // envia o pacote ao servidor
22
23         DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length); // cria o pacote de
24                                                                                                     // recebimento
25         clientSocket.receive(receivePacket); // recebe o pacote do servidor (o mesmo pacote que ele enviou mas em
26                                                                                                     // maiusculo)
27         clientSocket.close();
28
29         System.out.println("FROM SERVER:" + new String(receivePacket.getData()));
30     }
31 }
```

Como o código está funcionando:

1. O Servidor fica em loop aguardando mensagens UDP na porta **9871**;
2. O Cliente envia uma mensagem para o servidor (que digitamos no terminal);
3. O Servidor coloca a mensagem em maiúsculas e retorna;
4. O Cliente exibe a resposta recebida.