Politécnico de Coimbra

Escola Superior de Tecnologia e Gestão

Licenciatura em Engenharia Informática

Sistemas Distribuídos

Ano Letivo 2021/22

Ficha 2

Elaborado em: 2022/03/04

Daniel Santos | Nº2019133865

Índice

List of Figures ii

1. Exercício 1 1

2. Exercício 2 3

List of Figures

FIGURA 1 - RECEIVERUDP 1
FIGURA 2 – SENDERUDP 1
FIGURA 3 - RECEIVERUDP DATAGRAMPACKET 1
FIGURA 4 - SENDERUDP DATAGRAMPACKET 2
FIGURA 5 - SERVIDORDAYTIME 3
FIGURA 6 – CLIENTEDAYTIME 3
FIGURA 7 – CLIENTEDAYTIME CÓDIGO 3
FIGURA 8 – SERVIDORDAYTIME CÓDIGO 4

1. Exercício 1

Execução do programa:

```
Introduza o porto: 5000
A receber...
IP: /127.0.0.1 Porto: 53453 Tamanho da mensagem(bytes):5 Mensagem: thgrf
IP: /127.0.0.1 Porto: 53454 Tamanho da mensagem(bytes):4 Mensagem: ola
IP: /127.0.0.1 Porto: 53455 Tamanho da mensagem(bytes):9 Mensagem: tudo bem?
```

Figura 1 - ReceiverUDP

```
Introduza o endereço destino: localhost
Introduza o porto destino: 5000
Introduza a mensagem que deseja enviar: thgrf
Introduza a mensagem que deseja enviar: ola
Introduza a mensagem que deseja enviar: tudo bem?
Introduza a mensagem que deseja enviar:
```

Figura 2 – SenderUDP

Como desenvolvi o código:

Criei duas classes: ReceiverUDP e SenderUDP.

Na ReceiverUDP temos que introduzir o porto que o programa irá receber a informação.

Na SenderUDP temos que introduzir o endereço de ip e porto destino e a mensagem que se deseja enviar para o ReceiverUDP.

O ReceiverUDP para receber packets usa o DatagramPacket para receber mensagens (inbuf).

```
DatagramPacket dp = new DatagramPacket(inbuf, inbuf.length);
DatagramSocket socket = new DatagramSocket(port);
```

Figura 3 - ReceiverUDP DatagramPacket

O SenderUDP para mandar packets usa o DatagramPacket para enviar mensagens (data, ip address, port).

```
byte[] data = s.getBytes();
if("sair".equals(s)){
    System.out.println("A sair...");
    System.exit(0);
}
DatagramPacket dp = new DatagramPacket(data, data.length, ia, port);
DatagramSocket socket = new DatagramSocket();
```

Figura 4 - SenderUDP DatagramPacket

2. Exercício 2

Execução do programa:

```
Introduza o endereço destino: localhost
04-03-2022 16:36:46
BUILD SUCCESSFUL (total time: 4 seconds)
```

Figura 5 - ServidorDaytime

```
A receber...
IP: /127.0.0.1 Porto: 63187
```

Figura 6 – ClienteDaytime

Como desenvolvi o código:

Primeiro comecei por criar o ClienteDaytime onde irei ter que introduzir o endereço ip do ServidorDaytime.

O ClienteDaytime começa por enviar um packet sem mensagem ao ServidorDaytime e depois fica à escuta até que o ServidorDaytime mande uma mensagem de tempo. Se o ServidorDaytime demorar a mais que 4 segundos a responder ao ClienteDaytime, este irá dar timeout e acaba o programa.

```
//enviar
InetAddress ia = InetAddress.getByName(inputString("Introduza o endereço destino: "));
int port = 13;
String s = "";
byte[] data = s.getBytes();
DatagramPacket dpSend = new DatagramPacket(data, data.length, ia, port);
DatagramSocket socket = new DatagramSocket();
socket.send(dpSend);

//receber
DatagramPacket dpReceive = new DatagramPacket(inbuf, inbuf.length);
try {
    socket.setSoTimeout(4000);
    socket.receive(dpReceive);
} catch (SocketTimeoutException e) {
    System.out.println("Sem Resposta...");
}

String conteudo = new String(dpReceive.getData(), 0, dpReceive.getLength());
System.out.println(conteudo);
```

Figura 7 – ClienteDaytime código

No ServidorDaytime o utilizador não precisa de introduzir nada visto que por default o protocolo Daytime (RFC 867) define o porto como 13.

No ServidorDaytime fez-se um ciclo infinito que primeiro recebe o packet do ClienteDaytime e através dessa mensagem o ServidorDaytime obtém o ip e porto do ClienteDaytime e retorna a mensagem com a data e hora do momento atual (usou-se a biblioteca: java.time.*).

```
int port = 13;
System.out.println("A receber...");

DatagramPacket dpReceive = new DatagramPacket(inbuf, inbuf.length);
DatagramSocket socket = new DatagramSocket(port);
while (true) {
    //receber
    socket.receive(dpReceive);
    String conteudo = new String(dpReceive.getData(), 0, dpReceive.getLength());
    System.out.println("IP: " + dpReceive.getAddress() + " Porto: " + dpReceive.getPort());

    //enviar
    LocalDateTime currentDateTime = LocalDateTime.now();
    DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy HH:mm:ss");
    String s = currentDateTime.format(formatter);
    byte[] data = s.getBytes();
    DatagramPacket dpSend = new DatagramPacket(data, data.length, dpReceive.getAddress(), dpReceive.getPort());
    socket.send(dpSend);
```

Figura 8 – ServidorDaytime código

Como ambos os programas têm que receber e enviar pacotes criei dois tipos de packets: dpSend e dpReceive.