

## Prática 1 - Sockets em Java

### 1. Sockets UDP

- Cliente

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class ClienteSimplesUDP{

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

        //DatagramSocket para o cliente
        DatagramSocket socketCliente = new DatagramSocket();

        InetAddress endereco = InetAddress.getByName("localhost");

        // entrada de dados do cliente
        Scanner sc = new Scanner(System.in);
        System.out.println("Digite uma mensagem >\n");
        String msg = sc.nextLine();

        // transformando em bytes
        byte[] bufferEnvio = msg.getBytes();

        //Enviar datagrama para destinatário na porta 5050.
        DatagramPacket datagramaEnvio = new DatagramPacket(bufferEnvio,
bufferEnvio.length, endereco, 5050);

        System.out.println("Cliente UDP "+endereco.getHostAddress()+" " +
socketCliente.getLocalPort() +
            " enviando mensagem > "+msg);

        //Envia o datagrama.
        socketCliente.send(datagramaEnvio);

        //Passa um recvBuffer e o tamanho do recvBuffer para receber a
mensagem.
        //Se o conteúdo da mensagem recebida for maior que o recvBuffer o
//excedente é descartado.
        byte [] bufferRecebimento = new byte[1024];

        DatagramPacket datagramaRecebimento = new
DatagramPacket(bufferRecebimento, bufferRecebimento.length);

        socketCliente.receive(datagramaRecebimento);

        bufferRecebimento = datagramaRecebimento.getData();

        System.out.println("Cliente UDP "+endereco.getHostAddress()+"
```

```

recebeu mensagem < "+ new String(bufferRecebimento));

        sc.close();

        //Fechando o socket.
        socketCliente.close();
    }
}

```

- Servidor

```

package udp.server;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;

public class ServidorSimplesUDP {

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

        //DatagramSocket servidor na porta 5050.
        DatagramSocket datagramSocket = new DatagramSocket(5050);
        System.out.println("Servidor UDP executando na porta " +
datagramSocket.getLocalPort());

        //Cria o datagrama para receber uma mensagem.
        byte [] bufferRecebimento = new byte[1024];

        DatagramPacket datagramaRecebimento = new
DatagramPacket(bufferRecebimento, bufferRecebimento.length);

        //Aguarda até o recebimento de uma mensagem.
        datagramSocket.receive(datagramaRecebimento);

        bufferRecebimento = datagramaRecebimento.getData();

        String mensagem = new String(bufferRecebimento);

        System.out.println("Servidor UDP recebeu a mensagem < "+ mensagem);
        System.out.println("Preparando uma resposta");

        // Enviando uma resposta
        String msgResposta = "reply:" +
ServidorSimplesUDP.inverteString(mensagem);
        System.out.println(msgResposta);

        byte[] bufferEnvio = msgResposta.getBytes();

        DatagramPacket pacoteEnvio = new DatagramPacket (bufferEnvio,
bufferEnvio.length, datagramaRecebimento.getAddress(),
datagramaRecebimento.getPort());

        datagramSocket.send(pacoteEnvio);

        System.out.println("Servidor UDP enviou resposta > "+ msgResposta);

        //Fechando o servidor.
        datagramSocket.close();
    }
}

```

```
public static String inverteString(String s){  
    String retorna = "";  
    s = s.trim();  
    int len = s.length();  
    for(int i=len; i > 0; i--){  
        if (i-1>=0) retorna = retorna+s.charAt(i-1);  
    }  
    return retorna;  
}
```

#### Exercícios:

1. Crie um projeto no Eclipse, codifique, execute e observe o funcionamento dessa aplicação cliente/servidor simples.
2. Crie uma aplicação onde o cliente envia uma mensagem ao servidor envia três mensagens ao cliente, contendo, seu nome, seu endereço e sua idade.
3. Crie uma aplicação onde o cliente envia um valor numérico ao servidor. Se o valor for par, o servidor multiplica-o por 10 e envia o retorno ao cliente. Se o valor for ímpar, o servidor multiplica-o por 11 e envia o retorno ao cliente. Se o valor for zero, o servidor envia o valor zero por extenso ao cliente.