Nome: Bruno Felix Dias - Matrícula: 428903
Nome: Paulo Ricardo da Silva Lopes - Matrícula: 385173

Questão 1

- Neste trabalho baixamos e testamos os códigos Cliente e Servidor para comunicação UDP e TCP via Socket.
  Inicialmente falaremos sobre o modelo UDP. No código do Cliente basicamente o que adicionamos foi um *"input"*, uma forma do usuário digitar algo e essa mensagem ir direto para o Servidor que retorna a mensagem logo em seguida. Essa mensagem vai através de um *array de Bytes* e a comunicação Cliente-Servidor não há conexão, logo há o risco da mensagem não ser recebida.
  Código Cliente UDP:

```
1  public class ClientUDP {
2          private static Scanner input = new Scanner(System.in);
3
4          public static void main(String[] args) {
5                  DatagramSocket aSocket;
6                  try {
7                          aSocket = new DatagramSocket();
8                          String data = input.nextLine();
9                          byte[] m = data.getBytes();
10                         InetAddress aHost = InetAddress.getByName("
                               localhost");
11                         int serverPort = 6789;
12                         DatagramPacket request = new DatagramPacket(m,
                               data.length(), aHost, serverPort);
13                         aSocket.send(request);
14                         byte[] buffer = new byte[1000];
15                         DatagramPacket reply = new DatagramPacket(buffer,
                               buffer.length);
16                         aSocket.receive(reply);
17                         System.out.println("Reply:" + new String(reply.
                               getData()));
18                 } catch (Exception e) {
19                         System.out.println("Erro: " + e);
20                 }
21         }
22 }
```

- Aqui temos o código do Servidor UDP:

```
1
2  public class ServidorUDP {
3          public static void main(String[] args) {
4                  DatagramSocket aSocket = null;
5                  try {
6                          aSocket = new DatagramSocket(6789);
7                          byte[] buffer = new byte[1000];
8                          while (true) {
9                                  DatagramPacket request = new
                                       DatagramPacket(buffer, buffer.length);
10                                 aSocket.receive(request);
```

```
11                                  DatagramPacket reply = new DatagramPacket(
                                        request.getData(), request.getLength(),
                                         request.getAddress(),
12                                                      request.getPort());
13                                  aSocket.send(reply);
14                          }
15                  } catch (Exception e) {
16                          System.out.println("Erro: " + e);
17
18                    }
```

- Basicamente o que foi modificado foi apenas a variável que recebe o *array de Bytes*.
  Também devemos notar a importância do tamanho do *Buffer*. Caso a mensagem exceda o tamanho do *Buffer*, parte da mensagem será perdida.

- Já aqui temos o modelo de comunicação TCP que é orientado a conexão, diferente do modelo UDP que é orientado a mensagens.
  Código Cliente TCP:

```
1
2  public class TCPClient {
3          public static void main (String args[]) {
4                  // arguments supply message and hostname
5                  Socket s = null;
6                  try{
7                          String tTCP = "Teste server TCP";
8                          int serverPort = 7896;
9                          s = new Socket("localhost", serverPort);
10                         DataInputStream in = new DataInputStream( s.
                               getInputStream());
11                         DataOutputStream out =new DataOutputStream( s.
                               getOutputStream());
12                         out.writeUTF(tTCP);             // UTF is a string
                               encoding see Sn. 4.4
13                         String data = in.readUTF();         // read a line
                               of data from the stream
14                         System.out.println("Received: "+ data) ;
15                 }catch (UnknownHostException e)......
```

- Modificamos também a variável que é responsável por enviar o fluxo de dados onde o servidor lê esses dados via UTF.
  Basicamente o que fizemos foi testar as comunicações dos modelos e fazer as modificações necessárias para testar suas funcionalidades.
  Código do Servidor TCP:

```
1
2  public class TCPServer {
3          public static void main (String args[]) {
4                  try{
5                          int serverPort = 7896; // the server port
6                          ServerSocket listenSocket = new ServerSocket(
                               serverPort);
7                          while(true) {
8                                  Socket clientSocket = listenSocket.accept
                                       ();
```

```
9                                   Connection c = new Connection(clientSocket
                                       );
10                          }
11                  } catch(IOException e) {System.out.println("Listen socket:
                        "+e.getMessage());}
12          }
13  }
14  class Connection extends Thread {
15          DataInputStream in;
16          DataOutputStream out;
17          Socket clientSocket;
18          public Connection (Socket aClientSocket) {
19                  try {
20                          clientSocket = aClientSocket;
21                          in = new DataInputStream( clientSocket.
                                getInputStream());
22                          out =new DataOutputStream( clientSocket.
                                getOutputStream());
23                          this.start();
24                  } catch(IOException e) .....
```

Referências

[GUJ] https://www.guj.com.br/ - acessado em 20 set. 2019.

[Coulouris et al. 2013] Coulouris, G., Dollimore, J., Kindberg, T., and Blair, G. (2013). *Sistemas Distribuídos - Conceitos e Projetos*. Bookman Companhia Editora LTDA, 5ª edition.

[Tanenbaum and Steen 2008] Tanenbaum, A. S. and Steen, M. V. (2008). *Sistemas Distribuídos - Princípios e Paradigmas*. Pearson Education do Brasil, 2ª edition.