

Atividade 01 – Respostas:

Servidor:

```
package com.java;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;

public class Servidor {

    // Servidor - ServerSocket
    public static void main(String[] args) throws IOException{
        try {
            ServerSocket server = new ServerSocket(50000);
            System.out.println("Server started !!! \n");
            boolean up = true;
            while (true) {
                Socket socket = server.accept();
                DataInputStream entrada = new DataInputStream(socket.getInputStream());
                DataOutputStream saida = new DataOutputStream(socket.getOutputStream());

                String cpf = entrada.readUTF();
                boolean valido = ValidaCPF.validaCpf(cpf);
                String resultado = valido ? "TRUE" : "FALSE";
                System.out.println("Resultado.....: " + valido);

                if (valido) {
                    System.out.println("CPF Valido\n");
                }else {
                    System.out.println("CPF Invalido\n");
                }

                saida.writeUTF(resultado);
                socket.close();

                if (!up) {
                    server.close();
                    System.out.println("\nServer closed !!!");
                }
            }
        } catch (IOException e) {

        }
    }
}
```

Cliente:

```
package com.java;

import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.Socket;

public class Cliente {

    // Cliente - Socket
    public static void main(String[] args) {

        System.out.println("Inicio...\n");
        System.out.println("-----");
        System.out.println("Client started !!!");
        System.out.println("-----\n");

        try {
            boolean up = true;
            while (true) {
                System.out.println("-----");
                Socket socket = new Socket("127.0.0.1", 50000);
                DataInputStream entrada = new DataInputStream(socket.getInputStream());
                DataOutputStream saida = new DataOutputStream(socket.getOutputStream());
                BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

                System.out.println("-----");
                System.out.println("Digite o CPF: ");
                saida.writeUTF(br.readLine());

                if (entrada.readUTF().equals("TRUE")) {
                    System.out.println("\nCPF Valido\n");
                } else {
                    System.out.println("\nCPF Invalido\n");
                }
                System.out.println("-----");

                if (!up) {
                    socket.close();
                    System.out.println("\nClient closed !!!");
                }
                System.out.println("-----");
            }
        } catch (IOException e) {
            e.getMessage();
        }
        System.out.println("\nFim...");
    }
}
```

ValidaCPF:

```
package com.java;

public class ValidaCPF {
    // metodo que valida o cpf
    public static boolean validaCpf(String cpf) {
        System.out.println("CPF.....: " + cpf);
        if (validString(cpf)) {
            Integer xsoma = getDigito(cpf);
            Integer xresto = (xsoma % 11);
            Integer x = (xresto == 0 || xresto == 1) ? 0 : 11-xresto;
            System.out.println("Penultimo Digito..: " + x);

            Integer ysoma = getDigito(cpf.substring(1,9)+x);
            Integer yresto = (ysoma % 11);
            Integer y = (yresto == 0 || yresto == 1) ? 0 : 11-yresto;
            System.out.println("Ultimo Digito.....: " + y);

            String valida = x.toString() + y.toString() ;
            System.out.println("Digitos validar...: " + valida);
            System.out.println("Digitos cpf.....: " + cpf.substring(9,11));
            return valida.equals(cpf.substring(9,11));
        }
        return false;
    }
    // valida string digitada
    private static boolean validString(String cpf) {
        return !cpf.isEmpty() && cpf.length() == 11 && cpf.matches("[0-9]+") && verificaIguais(cpf);
    }
    // verifica se todos os caracteres sao iguais
    private static boolean verificaIguais(String text) {
        String[] listaCarga= new String[11];
        int cont = 0;
        for (int i = 0; i < text.length(); i++) {
            listaCarga[i] = text.substring(i, i+1);
        }
        for (int i = 0; i < listaCarga.length; i++) {
            for (int j = 0; j < text.length(); j++) {
                if (i < 10 && listaCarga[i].equals(text.substring(i+1, i+2))) {cont++;};
            }
        }
        return cont == 110 ? false : true;
    }
    // get digitos finais
    private static int getDigito(String cpf) {
        int soma= 0;
        for (int j = 0; j < 10; j++) {
            if (10-j!=1) {
                soma += (Integer.parseInt(cpf.substring(0+j, 1+j)))*(10-j);
            }
        }
        return soma;
    }
}
```

Cenários de testes:

<pre>----- Client started !!! -----</pre>	<pre>----- Server started !!! -----</pre>
<pre>----- Digite o CPF: 59946777010 CPF Valido -----</pre>	<pre>CPF.....: 59946777010 Penultimo Digito..: 1 Ultimo Digito.....: 0 Digitos validar...: 10 Digitos cpf.....: 10 Resultado.....: true CPF Valido</pre>
<pre>----- Digite o CPF: 27283351094 CPF Valido -----</pre>	<pre>CPF.....: 27283351094 Penultimo Digito..: 9 Ultimo Digito.....: 4 Digitos validar...: 94 Digitos cpf.....: 94 Resultado.....: true CPF Valido</pre>
<pre>----- Digite o CPF: 04252308993 CPF Invalido -----</pre>	<pre>CPF.....: 04252308993 Penultimo Digito..: 9 Ultimo Digito.....: 2 Digitos validar...: 92 Digitos cpf.....: 93 Resultado.....: false CPF Invalido</pre>
<pre>----- Digite o CPF: 99999999999 CPF Invalido -----</pre>	<pre>CPF.....: 99999999999 Resultado.....: false CPF Invalido</pre>

<pre>----- ----- Digite o CPF: 12345678910 CPF Invalido ----- -----</pre>	<pre>CPF.....: 12345678910 Penultimo Digito..: 0 Ultimo Digito.....: 9 Digitos validar...: 09 Digitos cpf.....: 10 Resultado.....: false CPF Invalido</pre>
<pre>----- ----- Digite o CPF: 123 CPF Invalido ----- -----</pre>	<pre>CPF.....: 123 Resultado.....: false CPF Invalido</pre>
<pre>----- ----- Digite o CPF: abcdefghijkl CPF Invalido ----- -----</pre>	<pre>CPF.....: abcdefghijk Resultado.....: false CPF Invalido</pre>