

Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas

REDES DE COMPUTADORES 1

Primeiro Trabalho

Guilherme Marx Ferreira Tavares - 14.1.8006
Rafael Júnio Cota Cekiera - 14.2.5834
Professor - Theo Lins

João Monlevade
11 de novembro de 2017

Sumário

1	Introdução	1
2	Algoritmos	1
2.1	Servidor	1
2.2	Cliente	1
3	Apresentação e discussão dos resultados	2
3.1	Discussão dos resultados	10
4	Apêndice A	10
5	Bibliografia	13

1 Introdução

2 Algoritmos

Nesta seção serão descritos a implementação e o funcionamento o programa.

2.1 Servidor

- teste

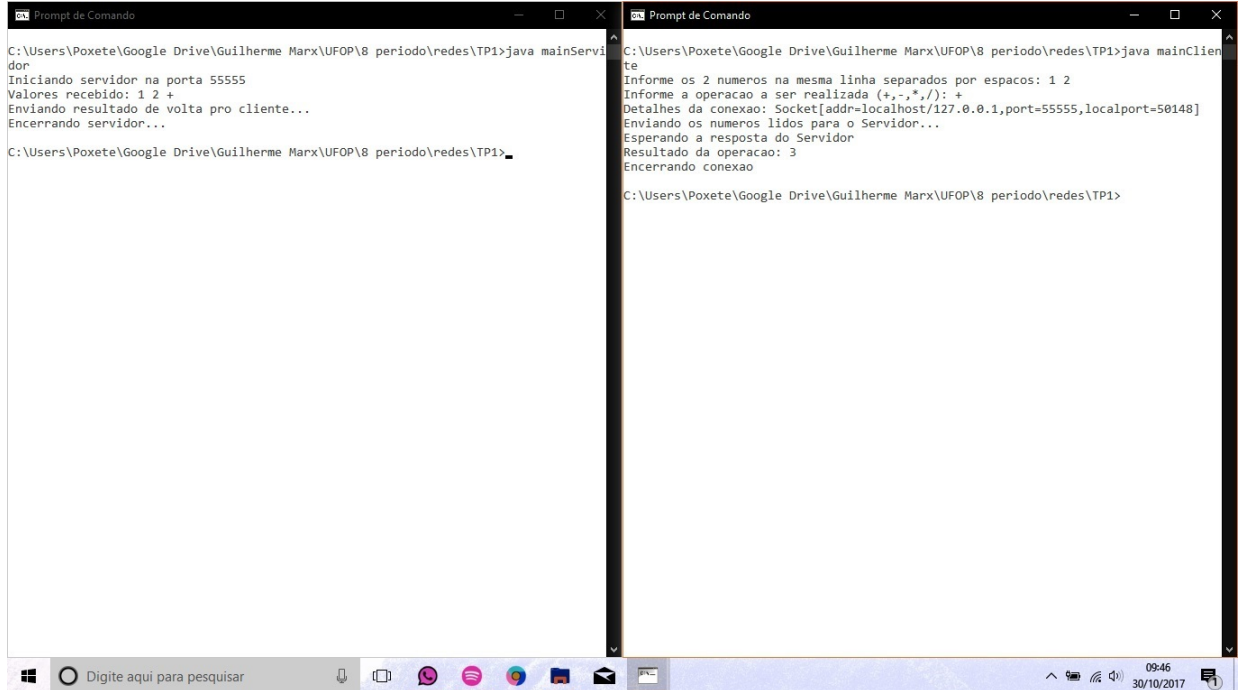
2.2 Cliente

- 111

3 Apresentação e discussão dos resultados

Foram realizado os testes de funcionamento do programa:

- Teste programa:



```
C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>java mainServidor
Iniciando servidor na porta 55555
Valores recebido: 1 2 +
Enviando resultado de volta pro cliente...
Encerrando servidor...

C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>

C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>java mainCliente
Informe os 2 numeros na mesma linha separados por espacos: 1 2
Informe a operacao a ser realizada (+,-,*,/): +
Detalhes da conexao: Socket[addr=localhost/127.0.0.1,port=55555,localport=50148]
Enviando os numeros lidos para o Servidor...
Esperando a resposta do Servidor
Resultado da operacao: 3
Encerrando conexao

C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>
```

Figura 1: Tela do teste

```
C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>java mainServidor
Iniciando servidor na porta 55555
Valores recebido: 10 2 *
Enviando resultado de volta pro cliente...
Encerrando servidor...
C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>

C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>java mainCliente
Informe os 2 numeros na mesma linha separados por espacos: 10 2
Informe a operacao a ser realizada (+,-,*,/): *
Detalhes da conexao: Socket[addr=localhost/127.0.0.1,port=55555,localport=50156]
Enviando os numeros lidos para o Servidor...
Esperando a resposta do Servidor
Resultado da operacao: 20
Encerrando conexao
C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>
```

Figura 2: Tela do teste

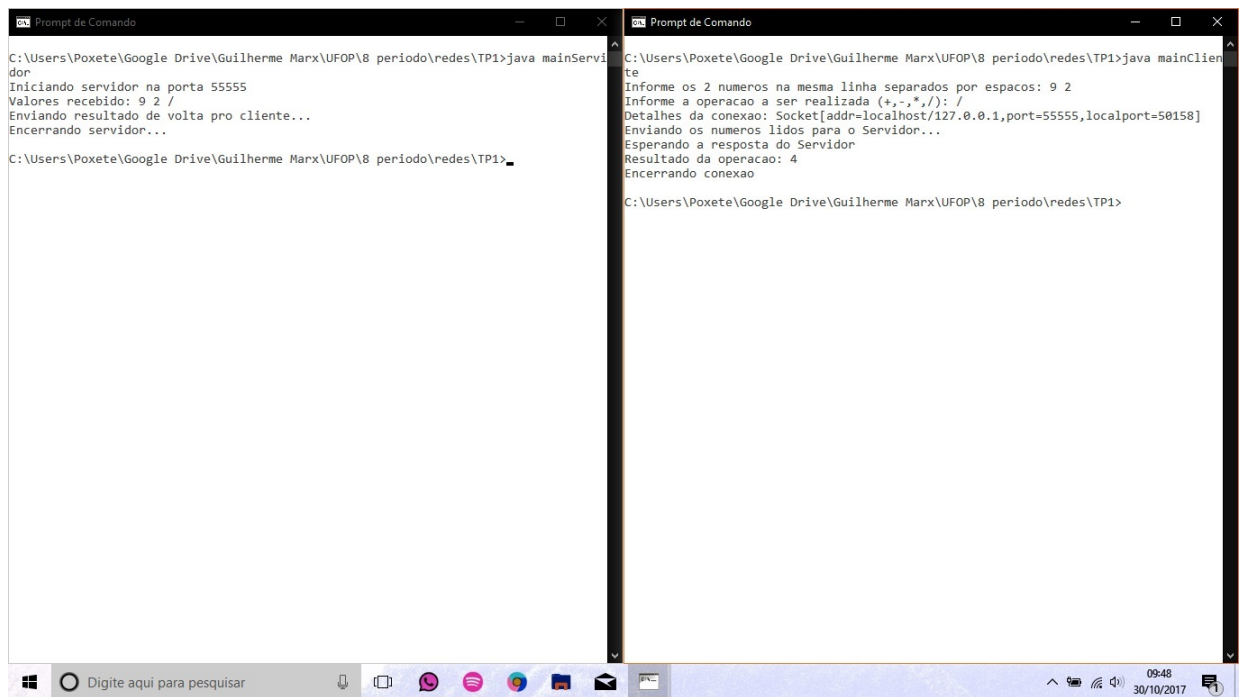


Figura 3: Tela do teste

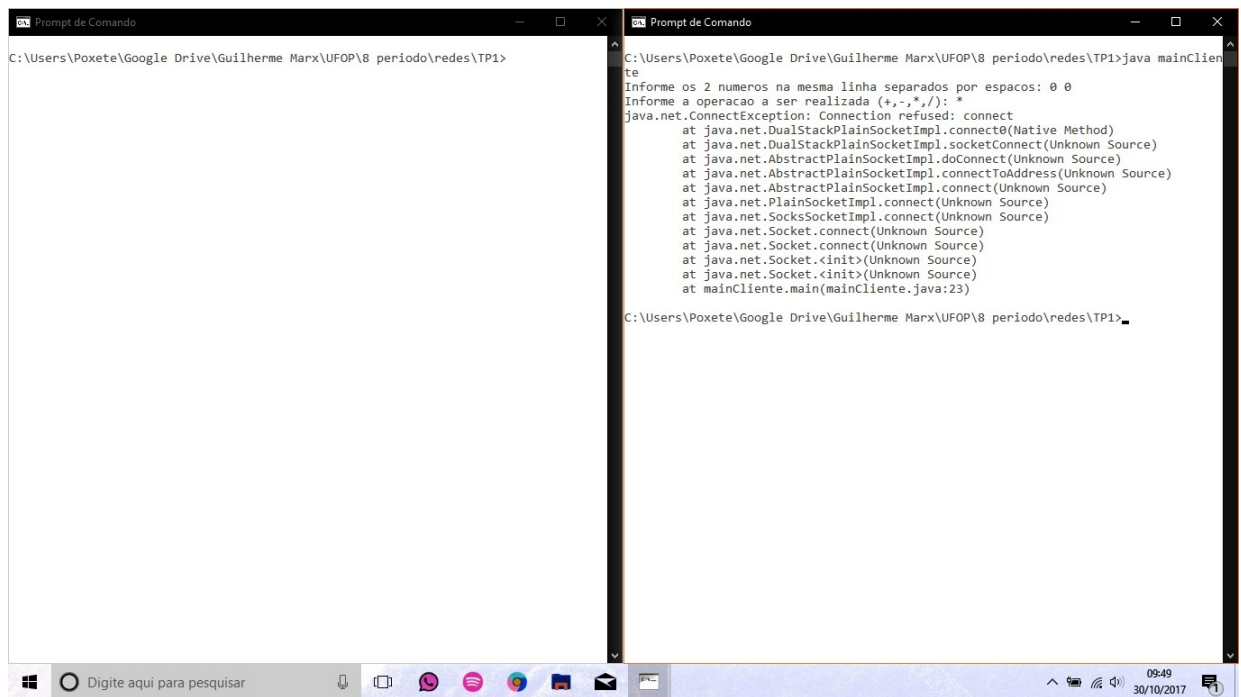


Figura 4: Tela do teste

```
C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>java mainServidor
Iniciando servidor na porta 55555
Valores recebido: 5 10 -
Enviando resultado de volta pro cliente...
Encerrando servidor...
C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>

C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>java mainCliente
Informe os 2 numeros na mesma linha separados por espacos: 5 10
Informe a operacao a ser realizada (+,-,*,/): -
Detalhes da conexao: Socket[addr=localhost/127.0.0.1,port=55555,localport=50161]
Enviando os numeros lidos para o Servidor...
Esperando a resposta do Servidor
Resultado da operacao: -5
Encerrando conexao
C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>
```

Figura 5: Tela do teste

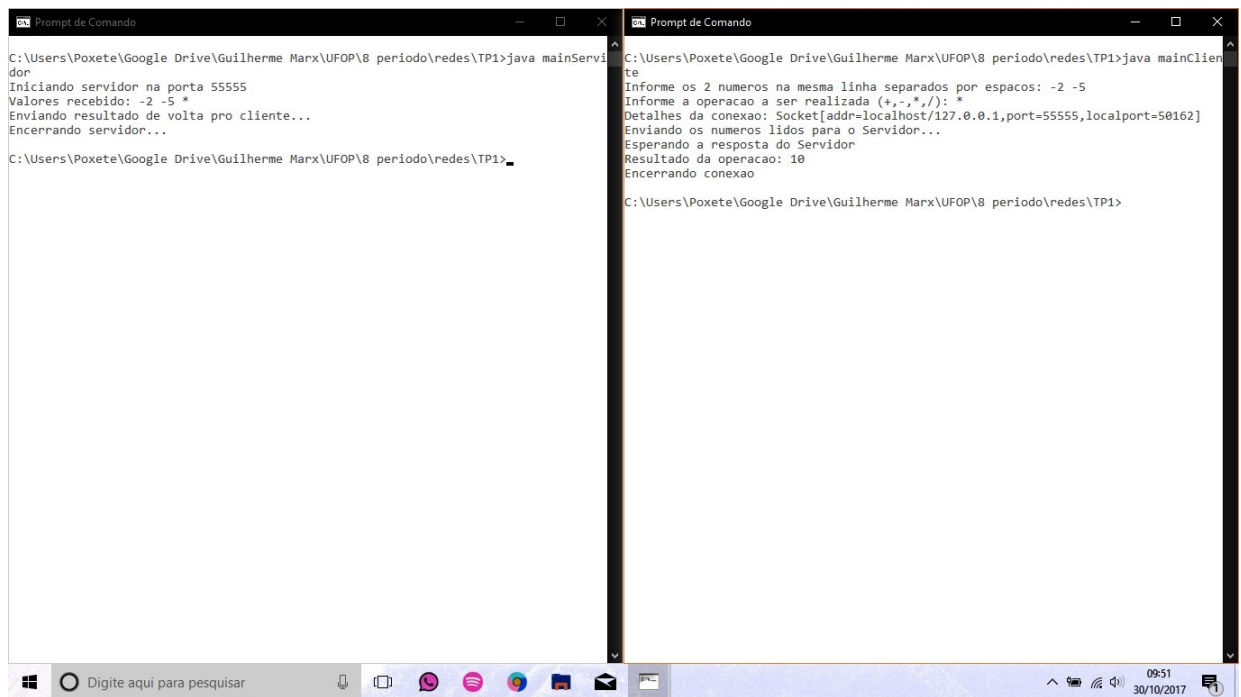


Figura 6: Tela do teste

```
C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>java mainServidor
Iniciando servidor na porta 5555
Valores recebido: 2 0 /
Exception in thread "main" java.lang.ArithmeticException: / by zero
    at mainServidor.main(mainServidor.java:33)
C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>

C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>java mainCliente
Informe os 2 numeros na mesma linha separados por espacos: 2 0
Informe a operacao a ser realizada (+,-,*,/): /
Detalhes da conexao: Socket[addr=localhost/127.0.0.1,port=55555,localport=50163]
Enviando os numeros lidos para o Servidor...
Esperando a resposta do Servidor
java.net.SocketException: Connection reset
    at java.net.SocketInputStream.read(Unknown Source)
    at java.net.SocketInputStream.read(Unknown Source)
    at java.net.SocketInputStream.read(Unknown Source)
    at java.io.DataInputStream.readUnsignedShort(Unknown Source)
    at java.io.DataInputStream.readUTF(Unknown Source)
    at java.io.DataInputStream.readUTF(Unknown Source)
    at mainCliente.main(mainCliente.java:36)
C:\Users\Poxete\Google Drive\Guilherme Marx\UFOP\8 periodo\redes\TP1>
```

Figura 7: Tela do teste

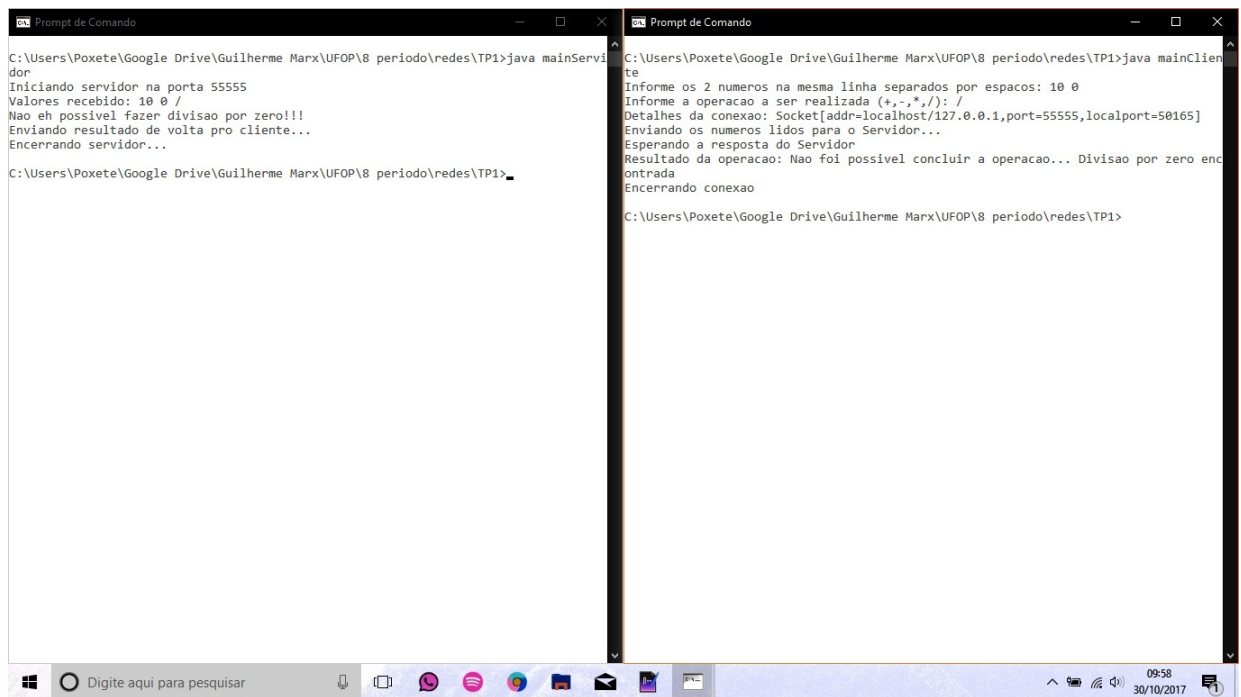


Figura 8: Tela do teste

3.1 Discussão dos resultados

4 Apêndice A

Programa 1: mainServidor.java

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

public class mainServidor {

    public static void main(String[] args) {

        System.out.print("Iniciando servidor na porta 55555\n");
        try {

            ServerSocket servidor = new ServerSocket(55555);
            Socket novaConexao = servidor.accept(); // aceita a conexao do
                                                    cliente

            DataOutputStream saida = new DataOutputStream(novaConexao.
                getOutputStream()); // Canal de saida
            DataInputStream entrada = new DataInputStream(novaConexao.
                getInputStream()); // Canal de entrada

            String valores = entrada.readUTF(); //recebe os valores enviados
                                                pelo cliente na forma "valor1 valor2 operacao"

            System.out.println("Valores recebido: " + valores);

            String[] n = valores.split(" "); //separa os valores recebidos
                                                cada um em uma string e a operacao em mais uma
            int valor1 = Integer.parseInt(n[0]), valor2 = Integer.parseInt(n
                [1]); //passa os valores nas strings para as variaveis
                                                inteiras

            int result = 0;
            boolean flag = false;
            switch(n[2]) { //le a operacao a ser realizada e a realiza
                case "+": result = valor1+valor2; break;
                case "-": result = valor1-valor2; break;
                case "*": result = valor1*valor2; break;
                case "/":
                    if(valor2 == 0)
                    {
                        System.out.println("Nao eh possivel fazer divisao por zero!!!
                            ");
                        flag = true;
                        break;
                    }
                result = valor1/valor2; break;
                default: break;
            }
        }
    }
}
```

```

String resultado = new String();
if(flag)
{
    resultado = "Nao foi possivel concluir a operacao... Divisao
                por zero encontrada";
} else
    resultado = String.format("%d",result);//salva na string
                resultado o valor calculado no switch

System.out.println("Enviando resultado de volta pro cliente...");
saida.writeUTF(resultado);//envia o resultado da operacao pro
                cliente

System.out.println("Encerrando servidor...");
novaConexao.close();//fecha a conexao com o cliente
servidor.close();//fecha o servidor

} catch (IOException e) {
    e.printStackTrace();
}

}

}

```

Programa 2: mainCliente.java

```

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.net.Socket;
import java.util.Scanner;

public class mainCliente {

    public static void main(String[] args) {

        String numero;
        String operacao;
        boolean valoresValidos = true;
        Scanner in = new Scanner(System.in);
        System.out.printf("Informe os 2 numeros na mesma linha separados
                        por espacos: ");
        numero = in.nextLine();

        System.out.printf("Informe a operacao a ser realizada (+,-,*,/): ");
        ;
        operacao = in.nextLine();

        String dado = String.format(numero + " " + operacao);

        try {
            Socket conexao = new Socket("localhost", 55555); // Conectando ao
                        servidor local na porta 5555
            System.out.println("Detalhes da conexao: " + conexao.toString());
        }
    }
}

```

```

DataOutputStream saida = new DataOutputStream(conexao.
    getOutputStream()); // Canal de saida
DataInputStream entrada = new DataInputStream(conexao.
    getInputStream()); // Canal de entrada

System.out.println("Enviando os numeros lidos para o Servidor..."
);

saida.writeUTF(dado); // Envia o numero lido para o servidor em
    formato de STRING codificada em UTF

System.out.println("Esperando a resposta do Servidor");

String resposta = entrada.readUTF(); // Recebe o resultado do
    Servidor em formato de STRING codificada em UTF

System.out.println("Resultado da operacao: " + resposta);

in.close(); // fecha o scanner
System.out.println("Encerrando conexao");
conexao.close(); // encerra a aplicacao do cliente
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

5 Bibliografia

[1] TENENBAUM, AARON M. Estruturas de dados usando C / AARON M. TENENBAUM, YEDIDYAH LANHSAM, MOSHE J. AUGENSTEIN; São Paulo : MAKRON Books, 1995.

[2] KUROSE, J. F. e ROSS, K. Redes de Computadores e a Internet / KUROSE, J. F. e ROSS, K; 5ª Ed., Pearson, 2010.

[3] HARVEY M. DEITEL & PAUL J. DEITEL. Java como programar / HARVEY M. DEITEL & PAUL J. DEITEL, K; 8ª Ed., Prentice Hall - Br, 2010.