



UNIVERSIDADE FEDERAL DE OURO PRETO
ICEA – INSTITUTO DE CIÊNCIA EXATAS E APLICADAS
CAMPUS JOÃO MONLEVADE
Engenharia Elétrica - Bacharelado

Trabalho pratico de redes

Alunos

Gustavo Alves Abreu14.2.8411

Talysson José Pereira14.2.8210

Outubro 2016

João Monlevade - MG



UNIVERSIDADE FEDERAL DE OURO PRETO
ICEA – INSTITUTO DE CIÊNCIA EXATAS E APLICADAS
CAMPUS JOÃO MONLEVADE
Engenharia Elétrica - Bacharelado

Sumário

Introdução.....	3
Implementação.....	4
Teste.....	7
Conclusão.....	9



1- Introdução

Neste trabalho prático, que tem a finalidade de aplicar o conceito de socket em redes de computadores, deve-se implementar uma comunicação entre processos situados em sistemas diferentes, por meio do mecanismo socket.

Um processo do cliente deve transmitir um número para o servidor, que representa a posição do número do Fibonacci que deve ser calculado e retornado para o resultado para o cliente.

O calculo funciona da seguinte forma:

- É iniciado o programa no servidor, para que o cliente pode ter acesso.
- O programa pede ao usuário para que digite o numero da posição do Fibonacci a ser retornado
- Após receber o número do cliente, o servidor abre a função Fibonacci recursiva e retorna o valor para o cliente.
- No cliente ele exibe a posição que foi digitada e o resultado do Fibonacci.



2- Implementação

A implementação teve como finalidade a comunicação entre duas máquinas utilizando sockets, para tal foi utilizado a linguagem Java.

Foram criadas duas classes uma para o servidor e outra para o cliente

As principais funções foram:

- Servesocket ela é responsável por esperar a conexão do cliente;
- Socket é o cliente propriamente dito e faz conexões com o servidor;
- ObjectOutputStream função capaz de enviar dados através de conexões;
- ObjectInputStream quando enviamos dados através do ObjectOutputStream recebemos dados com essa classe;
- JOptionPane função utilizada para imprimir dados na tela;
- SimpleDateFormat recebe uma data e formata sua saída;
- Process permite a execução de processos no Java.

As entradas e as saídas de dados:

As entradas necessárias são ip do servidor e a porta utilizada mas essas entradas foram passadas como parâmetros não inserir nenhum valor via teclado e o número digitado no cliente.

A saída é o valor do Fibonacci recebido do servidor.



UNIVERSIDADE FEDERAL DE OURO PRETO
ICEA – INSTITUTO DE CIÊNCIA EXATAS E APLICADAS
CAMPUS JOÃO MONLEVADE
Engenharia Elétrica - Bacharelado

Código do servidor

```
public class Main {

    private static int fibonacci(int numero) {
        if(numero == 0) return 0;
        else if(numero == 1) return 1;
        else return numero + fibonacci(numero -1);
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        System.out.print("Iniciando servidor na porta 55555");
        try {
            ServerSocket servidor = new ServerSocket(55555);
            Socket novaConexao = servidor.accept();

            DataOutputStream saidaDeDados = new
DataOutputStream(novaConexao.getOutputStream());
            DataInputStream entradaDeDados = new
DataInputStream(novaConexao.getInputStream());

            String parametroBruto = entradaDeDados.readUTF();
            int parametroTratado = Integer.parseInt(parametroBruto);

            int resultado = fibonacci(parametroTratado);

            System.out.println("O resultado da função de Fibonacci
para " + parametroBruto + " é " + String.valueOf(resultado));
            saidaDeDados.writeUTF(String.valueOf(resultado));

            System.out.println("Encerrando servidor...");
            novaConexao.close();
            servidor.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }

}
```

Código do cliente



UNIVERSIDADE FEDERAL DE OURO PRETO
ICEA – INSTITUTO DE CIÊNCIA EXATAS E APLICADAS
CAMPUS JOÃO MONLEVADE
Engenharia Elétrica - Bacharelado

```
public class Main {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        Scanner in = new Scanner(System.in);  
  
        System.out.println("Informe o número para ser calculado o  
Fibonacci: ");  
        String numero = in.nextLine();  
  
        try {  
            Socket conexao = new Socket("localhost", 55555); //  
Conectando ao servidor local na porta 55555  
            System.out.println("Conectado: " + conexao.toString());  
  
            DataOutputStream saidaDeDados = new  
DataOutputStream(conexao.getOutputStream()); // Canal de saida  
            DataInputStream entradaDeDados = new  
DataInputStream(conexao.getInputStream()); // Canal de entrada  
  
            saidaDeDados.writeUTF(numero); // Envia o número lido para  
o servidor em formato de STRING codificada em UTF  
            String resposta = entradaDeDados.readUTF(); // Recebe o  
resultado do Servidor em formato de STRING codificada em UTF  
  
            System.out.println("O resultado da função Fibonacci de " +  
numero + " é " + resposta + "\nCliente será encerrado.");  
            in.close();  
            conexao.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```



UNIVERSIDADE FEDERAL DE OURO PRETO
ICEA – INSTITUTO DE CIÊNCIA EXATAS E APLICADAS
CAMPUS JOÃO MONLEVADE
Engenharia Elétrica - Bacharelado

3- Testes

Iniciando o servidor

```
5 public class Main {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9
10        Scanner in = new Scanner(System.in);
11
12        System.out.println("Informe o número para ser calculado o Fibonacci:\n ");
13        String numero = in.nextLine();
14
15        try {
16            Socket conexao = new Socket("localhost", 8080); // Conectando ao servidor local na porta 5555
17            System.out.println("Conectado: " + conexao.toString());
18
19            DataOutputStream saidaDeDados = new DataOutputStream(conexao.getOutputStream()); // Canal de saída
20            DataInputStream entradaDeDados = new DataInputStream(conexao.getInputStream()); // Canal de entrada
21
22            saidaDeDados.writeUTF(numero); // Envia o número lido para o servidor em formato de STRING codificada em UTF
23            String resposta = entradaDeDados.readUTF(); // Recebe o resultado do Servidor em formato de STRING codificada em UTF
24
25            System.out.println("O resultado da função Fibonacci de " + numero + " é " + resposta + "\nCliente será encerrado.");
26        }
27    }
28 }
```

Problems | Javadoc | Declaration | Console

Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (24 de out de 2016 22:02:24)

Iniciando servidor na porta 5555

Incluindo a variável

```
5 public class Main {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9
10        Scanner in = new Scanner(System.in);
11
12        System.out.println("Informe o número para ser calculado o Fibonacci:\n ");
13        String numero = in.nextLine();
14
15        try {
16            Socket conexao = new Socket("localhost", 8080); // Conectando ao servidor local na porta 5555
17            System.out.println("Conectado: " + conexao.toString());
18
19            DataOutputStream saidaDeDados = new DataOutputStream(conexao.getOutputStream()); // Canal de saída
20            DataInputStream entradaDeDados = new DataInputStream(conexao.getInputStream()); // Canal de entrada
21
22            saidaDeDados.writeUTF(numero); // Envia o número lido para o servidor em formato de STRING codificada em UTF
23            String resposta = entradaDeDados.readUTF(); // Recebe o resultado do Servidor em formato de STRING codificada em UTF
24
25            System.out.println("O resultado da função Fibonacci de " + numero + " é " + resposta + "\nCliente será encerrado.");
26        }
27    }
28 }
```

Problems | Javadoc | Declaration | Console

Main [Java Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (24 de out de 2016 22:02:50)

Informe o número para ser calculado o Fibonacci:



UNIVERSIDADE FEDERAL DE OURO PRETO
ICEA – INSTITUTO DE CIÊNCIA EXATAS E APLICADAS
CAMPUS JOÃO MONLEVADE
Engenharia Elétrica - Bacharelado

Resultado

The screenshot shows the Eclipse IDE with a Java project named 'Trab Redes'. The 'Main.java' file is open, displaying a Fibonacci sequence calculator. The code includes a recursive function 'fibonacci' and a 'main' method that starts a server on port 8080, accepts a connection, reads a number from the client, calculates the Fibonacci value, and sends the result back to the client.

```
11     else if(numero == 2) return 1;
12     else return fibonacci(numero - 1) + fibonacci(numero - 2);
13 }
14
15 public static void main(String[] args) {
16     // TODO Auto-generated method stub
17
18     System.out.print("Iniciando servidor na porta 5555\n");
19     try {
20         ServerSocket servidor = new ServerSocket(8080);
21         Socket novaConexao = servidor.accept();
22
23         DataOutputStream saidaDeDados = new DataOutputStream(novaConexao.getOutputStream()); // Canal de saída
24         DataInputStream entradaDeDados = new DataInputStream(novaConexao.getInputStream()); // Canal de entrada
25
26         String parametroBruto = entradaDeDados.readUTF();
27         int parametroTratado = Integer.parseInt(parametroBruto);
28
29         int resultado = fibonacci(parametroTratado);
30
31         System.out.println("O resultado da função de Fibonacci para " + parametroBruto + " é " + String.valueOf(resultado));
32         saidaDeDados.writeUTF(String.valueOf(resultado));
33     } catch (IOException e) {
34         e.printStackTrace();
35     }
36 }
```

The console output shows the following messages:

```
<terminated> Main [Java Application] C:\Program Files\Java\jdk1.8.0_111\bin\javaw.exe (24 de out de 2016 22:05:09)
Informe o número para ser calculado o Fibonacci:
9
Conectado: Socket[addr=localhost/127.0.0.1,port=8080,localport=49629]
O resultado da função Fibonacci de 9 é 21
Cliente será encerrado.
```




UNIVERSIDADE FEDERAL DE OURO PRETO
ICEA – INSTITUTO DE CIÊNCIA EXATAS E APLICADAS
CAMPUS JOÃO MONLEVADE
Engenharia Elétrica - Bacharelado

4- Conclusão

Após a realização de alguns testes, percebeu-se o pleno funcionamento do programa implementado para a situação em questão.

Observou-se também o sucesso entre a comunicação entre o cliente o servidor, onde utilizou-se a porta 55555. Através desta porta que foi realizada o calculo do número Fibonacci.

Ao realizar a implementação, teve como principal dificuldade, a função para a estabelecer conexão entre o cliente e o servidor uma vez que precisa iniciar o cliente antes do servidor.

Contudo, o trabalho prático foi realizado com sucesso.