



Nome: Bruno Felix Dias - Matrícula: 428903

Nome: Paulo Ricardo da Silva Lopes - Matrícula: 385173

## Questão 2

- Nesta exercício nós, tendo em vista que já entendemos o funcionamento do modelo UDP, criamos uma classe Calculadora, que faz as operações básicas como somar, subtrair, multiplicar e dividir, e tratamos o *array de Bytes* enviado pelo processo Cliente no processo Servidor.

(Servidor) Código:

```
1 String s = new String(buffer);
2 char[] cArray = s.toCharArray();
3 char op = 'f';
4 Double n1 = 0.0;
5 Double n2 = 0.0;
```

- Basicamente o que fizemos foi tratar esse *array de Bytes* e transforma-lo em uma *String* e depois dividimos essa *String* e armazenamos em um *array de Char*. Feito todo esse procedimento, criamos um *laço* para percorrermos nosso *array* e encontrar nosso operando e armazena-lo em um único tipo primitivo *char*.

Após identificar e armazenar nosso operando nós identificamos e guardamos a *substring* antes do operador como nosso primeiro operador e a *substring* depois do operador como nosso segundo operando.

(Servidor) Código:

```
1 for (int i = 0; i < cArray.length; i++) {
2     if (cArray[i] == '+' || cArray[i] == '-' || cArray[i] == '*' ||
3         cArray[i] == '/') {
4         op = cArray[i];
5         String s1 = s.substring(0, s.indexOf(cArray[i])); //captura
6             o primeiro operando antes da opera o
7         String s2 = s.substring(s.indexOf(cArray[i]) + 1, cArray.
8             length); //captura o segundo operando antes da
9             opera o
10        n1 = Double.parseDouble(s1); //casting do primeiro operando
11        n2 = Double.parseDouble(s2); ///casting do segundo
12        operando
13        break;
14    }
15 }
```

- Depois apenas criamos um *switch* que faria a operação dependo de qual foi o operador identificado.

(Servidor) Código:

```
1 switch (op) {
2     case '+':
3         result = calc.soma(n1, n2);
4         resposta = Double.toString(result);
5         break;
6     case '-':
7         result = calc.subtracao(n1, n2);
8         resposta = result.toString();
9 }
```



```
9         break;
10     case '*':
11         result = calc.multiplic(n1, n2);
12         resposta = String.valueOf(result);
13         break;
14     case '/':
15         if (n2 == 0.0)
16             resposta = "Operacao invalida! Impossivel dividir por zero";
17         else {
18             result = calc.divisao(n1, n2);
19             resposta = result.toString();
20         }
21         break;
22     default:
23         //resposta = new String(buffer);
24         resposta = "ERRO! OPERACAO INVALIDA.";
25         break;
26 }
27 }
```

## 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.