



Universidade Federal do Ceará
Campus de Quixadá

Sistemas Distribuídos

Trabalho 1 – Prática com Sockets

Dupla: Bárbara Stéphanie Neves e Lucas Benjamim
Professor: Carlos Bruno Pereira Bezerra

**Setembro
2018**



Universidade Federal do Ceará
Campus de Quixadá

Sistemas Distribuídos

Questão 2

Relatório da disciplina de Sistemas Distribuídos que consiste em descrever os resultados da **questão 2** do **Trabalho 1 – Prática com Sockets**.

Dupla: Bárbara Stéphanie Neves e Lucas Benjamim

Matrículas: 388713 e 388717

Professor: Carlos Bruno Pereira Bezerra

Curso: Ciência da Computação

Setembro
2018

Conteúdo

1	Apresentação	1
2	Questão 2	1

1 Apresentação

Este relatório consiste em descrever os resultados obtidos ao responder a questão 2 do Trabalho 1 – Prática com Sockets da disciplina. O objetivo do trabalho é praticar a implementação de Sockets, tanto TCP quando UDP. O trabalho foi desenvolvido em Java.

2 Questão 2

Esta questão pede para implementar uma calculadora remota utilizando UDP que execute as funções básicas: adição, subtração, multiplicação e divisão. Também devemos mostrar o formato e o tipo das mensagens: se é uma requisição e se é uma resposta.

Para fazer esta questão tivemos 3 impasses. O primeiro deles se tratou em converter *string* em *buffer* e vice-versa para a manipulação dos dados. O segundo impasse foi tratar a *string* para conseguirmos efetuar as operações aritméticas. E o terceiro foi acerca de como tratar a resposta, já que, por exemplo, se a operação fosse uma multiplicação com números grandes, quando o cliente recebesse a resposta poderia estourar o *buffer* e mostrar o resultado incorreto. Analogamente, se a resposta for menor que o *buffer*, apesar dele imprimir a resposta correta, após a resposta existirão alguns caracteres que não simbolizam a nossa resposta.

Para a solução do primeiro impasse, utilizamos a função **getBytes()** da classe **String** para transformar a *string* em *buffer*. Para a ação contrária, instanciamos uma nova *string* passando os parâmetros dos dados do *buffer* do **offset**, que deve começar a partir de 0, e o tamanho do *buffer*.

Já para o segundo problema, utilizamos a função **split()** da classe **String** (dica que foi dada na questão para quem usasse Java).

Para resolvermos o terceiro impasse, instanciamos um novo *buffer* na classe **Client** contendo o dobro do tamanho do *buffer* que foi passado para o servidor. Dessa forma, resolvemos o problema de estouro. E, para resolver os problemas dos caracteres indesejáveis, na classe **Server** enviamos uma resposta com um **token** ao final do nosso resultado e tratamos esse erro na classe **Client**: utilizamos a função **split()** novamente passando o **token** como parâmetro, resolvendo o problema e extraindo o resultado de forma correta.