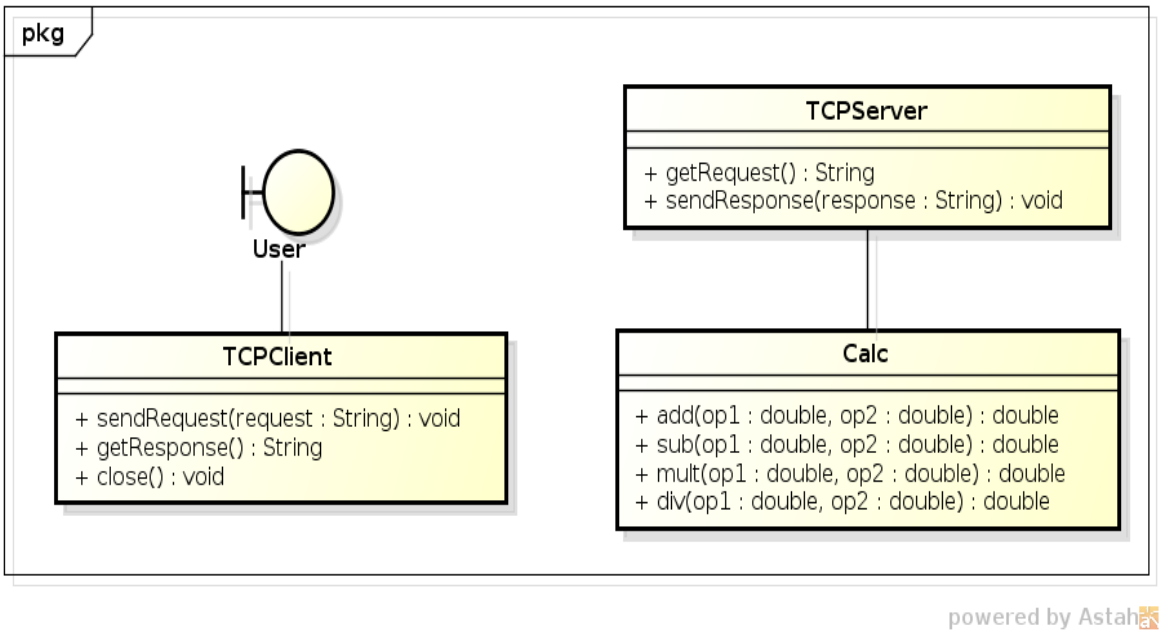
Universidade Federal do Ceará (UFC) - Campus Quixadá
Disciplina: Sistemas Distribuídos - Período 2021:2
Prof. Marcos Dantas Ortiz - mdo@ufc.br
Lista prática – 1 – complemento - entrega – 09/11/21

Questão 1 – Adicione um serviço simples de sua escolha ao processo servidor. Quais modificações são necessárias para oferecer **dois serviços no mesmo processo**? Compare essa solução com a criação de um processo servidor para cada serviço.

Questão 2 - Continuando a evolução da arquitetura Cliente-Servidor vista na Lista Prática - 1, devemos retirar a interação com o usuário da classe TCPClient e o serviço da classe TCPServer. Dessa forma, os objetos de TCPClient e TCPServer tornam-se c**oesos com o único objetivo de prover comunicação** (estabelecimento de conexão e trocas de mensagens através dos fluxos de entrada (IN) e saída (OUT)).



Questão 3: escreva um código de teste de carga para ambos servidores TCP:

- ***multithread*** (múltiplos clientes simultâneos)
- ***singlethread*** (atende um cliente por vez. A abertura de conexão (accept()) está na mesma *thread* que trata a requisição).

O código de teste deve gerar **200 clientes (*threads*) simultâneos**. Cada cliente fará uma única requisição ao servidor e encerrará. Faça uma rodada para o servidor *multithread* e outra para o *singlethread*. **Compare os tempos de execução.**

Não faça interação com o usuário no cliente (entrada por teclado). Todos as requisições podem ser iguais e definidas no próprio código: exemplo - "ADD;10;11".

Como a calculadora não oferece carga suficiente para a *thread* perder a CPU antes de terminar sua execução, adicione uma pausa no próprio código. Em java, por exemplo: **Thread.sleep(100)** ("a ***thread*** será suspensa por 100 milissegundos"). Dessa forma, o servidor *singlethread* vai levar pelo menos 200 x 100 milissegundos para tratar todas as 200 requisições.

Dica: **evite mandar mensagem de saída para o console pois ele será uma gargalo para as *threads* concorrentes** (println(), print(), System.out.println()).

Questão 4 – Transforme os Objetos que fornecem o serviço (calculadora, por exemplo) em ***Singletons***

- ***java:***https://www.devmedia.com.br/padrao-de-projeto-singleton-em-java/26392
- ***python:*** https://refactoring.guru/pt-br/design-patterns/singleton/python/example