

Universidade Federal de Campina Grande  
Departamento de Sistemas e Computação  
Disciplina: Programação concorrente - 2019.1  
Professor: Thiago Emmanuel Pereira  
Aluno: \_\_\_\_\_

(As respostas, tanto para as questões teóricas quanto para as práticas devem ser disponibilizadas em um repositório privado do github compartilhado com o professor. A avaliação considerará a defesa oral das soluções)

1. Considere um programa com duas funções principais indicadas abaixo:

***int gateway(int num\_replicas)***

***int request()***

A função **request** deve sortear um número aleatório entre 1 e 30, dormir por uma quantidade de segundos dada pelo número aleatório sorteado, e retornar o valor do número. Por sua vez, a função **gateway** deve iniciar **num\_replicas** goroutines, e cada thread goroutines a função **request**.

- a) Escreva uma versão do programa em que a função **gateway** retorna o valor da primeira goroutine que termine a execução da função **request** (sem que seja necessário esperar as demais goroutines terminem).
- b) Escreva uma versão do programa em que a função **gateway** retorna a soma dos valores retornados na execução da função **request** por todas as réplicas criadas.

obs 1) Pequenas mudanças na API indicada podem ser realizadas, para que seja possível adaptar o código para a linguagem indicada. Contacte o professor caso queira realizar alguma mudança.

2. Considere novas versões para os programas acima de modo que:
  - a. A função **gateway** retorne -1 caso nenhuma goroutine tenha terminado antes de 8 segundos nos programas da questão **3a**.
  - b. A função **gateway** retorne -1 caso alguma goroutine ainda esteja executando após 16 segundos nos programas da questão **3b**.
3. Avalie experimentalmente quão leve goroutines em Go são em relação à threads em Java. Em particular, você deve avaliar o consumo de memória de programas equivalentes nas respectivas linguagens.