

## LISTA 2 – Respostas

OBS.: Professor, como fiquei em dúvida sobre o 'retorno' da função gateway, fiz duas versões de cada questão: uma com prints fazendo o papel de return e outra apenas com os returns realmente.

1. No diretório 'versão 1', a questão 1 possui prints para melhor acompanhamento da execução do código. No diretório 'versão 2', não possui prints.  
Na letra a, a função gateway inicia num\_replicas goroutines da função request. E assim, que a primeira execução é finalizada (após o primeiro *first<-chosen*), a função gateway retorna seu valor.  
Na letra b, a função gateway inicia num\_replicas goroutines da função request. E logo em seguida aguarda o valor de cada uma dessas num\_replicas goroutines, somando em um valor total. Após, retorna o valor da soma total.
2. No diretório 'versão 1', a questão 2 possui prints para melhor acompanhamento da execução do código. No diretório 'versão 2', não possui prints.  
Na letra a, a função gateway inicia num\_replicas goroutines da função request. Logo em seguida, inicia uma goroutine para uma terceira função contador() que retorna '-1' após 8 segundos. E assim, a primeira goroutine a ser finalizada (uma das request() ou a contador()), terá seu valor retornado.  
  
Na letra b, a função gateway inicia num\_replicas goroutines da função request. Logo em seguida, inicia uma goroutine para uma terceira função contador() que retorna '-1' após 16 segundos. Ao decorrer da execução das goroutines, um somatório vai recebendo os valores das requests() encerradas. Caso o valor recebido seja '-1', sabe-se que o limite de 16 segundos foi alcançado, fazendo a função gateway retornar '-1'. Ao contrário, o somatório continua recebendo os valores das demais goroutines requests(), até a última, retornando o valor da soma total.
3. Não foi feita.