

# Lista de Exercícios 7 (Caps. 19 e 20) – INF05008

Siga as instruções sobre elaboração de exercícios de INF05008. Adicionalmente, esta lista tem 2 requisitos:

- Em nenhuma questão podem ser usadas estruturas condicionais.
- Você deve usar funções de alta-ordem em todas as questões (exceto no item (a) da questão 2).

1. Considere as definições de dados e os programas do template (questão 1). Reescreva o código das funções dos itens (A) a (E) usando as funções `filter`, `map` e/ou `foldl` (não é permitido usar outras funções de alta-ordem, somente estas). Lembre que o código das novas funções **não pode ter** nenhuma expressão condicional (`cond` ou qualquer outra expressão condicional, por exemplo `if`), mas deve preservar o comportamento da função, ou seja, quem chama a função não deve notar nenhuma diferença entre as duas versões, são apenas duas formas diferentes de implementar a mesma função. As novas funções devem ter os mesmos nomes das originais, então apague ou comente as funções originais. Inclua pelo menos 4 testes para cada uma das funções (A) e (E), sendo que para a funções que geram imagens, os testes devem ser como os da função `desenha`, ou seja, indicar em código (e não através de imagens) o desenho resultante da execução da função, sendo que este código só pode conter as seguintes funções pré-definidas de desenho do Racket: `triangle`, `ellipse`, `star`, `beside`, `empty-image`.
2. Considere as definições dos tipos de dados a seguir para fazer as questões (a) a (f):

```
(define-struct aplicativo (nome tamanho aval rede?))  
;; Um elemento de Aplicativo é  
;;   (make-aplicativo n t a r), onde  
;;   n : String, é o nome do aplicativo  
;;   t : Número, é a o tamanho que o aplicativo ocupa na memória  
;;   a : Número, é a nota de avaliação do aplicativo (entre 0 e 5)  
;;   r : Booleano, indica se o aplicativo precisa ou não de rede para executar (se for #t precisa de rede)  
  
;; Uma ListaDeAplicativos é  
;; 1. empty, ou  
;; 2. (cons a la), onde  
;;    a : Aplicativo  
;;    la: ListaDeAplicativos
```

- (a) Defina 3 funções quaisquer que, dado um aplicativo, verificam se este aplicativo atende ou não algum critério. Você pode escolher os critérios que quiser (por exemplo, se o tamanho do aplicativo é maior que 100, se a nota de avaliação é menor que 4, se o aplicativo precisa de rede, ...). O resultado deve ser um valor booleano.
- (b) Construa uma função chamada `filtra-otimos` que, dada uma lista de aplicativos, devolve os nomes dos aplicativos que tem nota de avaliação maior ou igual a 4,5.
- (c) Desenvolva uma função chamada `filtra-otimos-sem-rede` que, dada uma lista de aplicativos, devolve os nomes dos aplicativos que tem nota de avaliação maior ou igual a 4,5 e funcionam mesmo não conectados à rede.
- (d) Desenvolva a função `nota-media-aplicativos` que, dada uma lista de aplicativos, devolve a média (aritmética) de avaliação dos aplicativos da lista. Você pode usar a função `length` que, dada uma lista, diz quantos elementos a lista possui. Assuma que a lista de entrada nunca está vazia.
- (e) Ao invés de construir uma função para filtrar a lista com cada critério diferente, é mais interessante construir uma função genérica que recebe como um dos argumentos o critério de seleção. Construa a função `filtra-lista-aplicativos` que, dados uma lista de aplicativos e um critério de seleção, filtra esta lista de acordo com este critério. Lembre que o critério deve ser uma função, portanto a função `filtra-lista-aplicativos` é uma função de alta-ordem. Faça pelo menos 3 testes usando critérios diferentes.
- (f) *Desafio – nota extra: Construa a função `filtra-tamanho` que, dada uma lista de aplicativos e um tamanho, gera a lista de nomes de aplicativos que ocupam no máximo este tamanho. Note que, como o tamanho é um argumento da função e ele deve ser usado na filtragem, é necessário criar uma função de critério em tempo de execução (usando expressões lambda).*