

Exercícios sobre Funções de Alta Ordem

prof. André Rauber Du Bois

Universidade Federal de Pelotas
<http://sites.google.com/site/haskellufpel/>
dubois@inf.ufpel.edu.br

1 Questionário

1. Defina novamente a função `total`, que calcula o total de vendas de várias semanas. Agora a função `total` deve receber como argumento a função que determina as vendas de uma semana:

```
total :: (Int -> Int) -> Int -> Int
```

2. Defina a função `foldInt`

```
foldInt :: (Int -> Int -> Int) -> [Int] -> Int
```

```
soma :: Int -> Int -> Int
```

```
soma x y = x + y
```

```
> foldInt soma [1,2,3]
```

```
6
```

3. Testar a função `foldInt` com mais dois exemplos
4. Defina a função

```
filterString :: (Char -> Bool) -> [Char] -> [Char]
```

```
naoEspaco :: Char -> Bool
```

```
naoEspaco x = x /= ' '
```

```
> filterString naoEspaco "Andre Du Bois "
```

```
"AndreDuBois"
```

5. Testar `filterString` com mais dois exemplos
6. Usando as funções de alta ordem definidas nos exercícios anteriores (incluindo o `mapInt`), defina uma função que devolva a soma do quadrado dos números em uma lista `l`
7. Defina uma função que testa se os valores gerados por uma função `f` aplicada a valores de 0 até `n` são maiores que zero

8. Defina uma função `duasVezes` que recebe uma função de Inteiros para Inteiros, um Inteiro e aplique a função duas vezes ao argumento. Ex:

```
Hugs> duasVezes times2 2  
8
```

9. Dê o tipo e defina a função `inter`, onde

```
inter 3 f x = f(f (f x))
```