

# Exercícios sobre List Comprehensions

prof. André Rauber Du Bois

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

## 1 Questionário

1. Definir a função `filter` usando LC.
2. Definir a função `map` usando LC.
3. Definir a função

```
removeEspacos :: String -> String
```

4. Definir a função

```
sings :: [[a]] -> [a]
```

```
> sings [[1], [], [2,3,4], [9], [3,4], [6], [7,7,7]]  
[1,9,6]
```

5. Definir a função

```
matches :: Int -> [Int] -> [Int]
```

que retorna as ocorrências de um número em uma lista

```
> matches 1 [1,2,44,1,33,2,7,1]  
[1,1,1]  
> matches 1 [3,2,8,6]  
[]
```

6. Usando a função `matches` definir a função `elemento`

```
> elemento 1 [1,2,44,1,33,2,7,1]  
True  
> matches 1 [3,2,8,6]  
False
```

7. Definir a função

```
divisores :: Integer -> [ Integer]
```

```
> divisores 12  
[1,2,3,4,6,12]
```

8. Definir a função `isPrime` usando divisores

9. Dada a função `quickSort`

```
quickSort :: Ord a => [a] -> [a]
```

```
quickSort (a:l) = quickSort (menores a l) ++ [a] ++ quickSort (maiores a l)
```

Definir as funções `menores` e `maiores` usando LC