

# Aula de Revisão

## Programação Funcional

Maycon Amaro

### Introdução

Realizaremos juntos alguns exercícios envolvendo listas, tipos de dados algébricos e funções de ordem superior. Durante a aula e também ao final, eventuais dúvidas podem ser levantadas para esclarecimento. Utilizaremos o VSCode com a extensão para apenas destacar a sintaxe de Haskell e o Stack em linha de comando para compilação e teste manual. A diretiva para mostrar avisos estará ativa.

```
{-# OPTIONS_GHC -Wall #-}
```

```
module Revisao where
```

### Exercícios

Primeiro vamos implementar um *flattening* de listas. Dada uma lista de listas de um determinado tipo, vamos concatenar as sub-listas para formar uma lista de uma dimensão só. Faremos de várias formas, mas primeiro tentaremos com recursão.

```
exemploFlat :: [[Int]]
exemploFlat = [[11, 4, 8, 15, 22], [1, 3, 5], [7, 2, 9, 13]]
```

```
flatRec :: [[a]] -> [a]
flatRec = undefined
```

Agora vamos usar funções de ordem superior.

```
flat :: [[a]] -> [a]
flat = undefined
```

E ainda em estilo *point-free* =D

```
flat' :: [[a]] -> [a]
flat' = undefined
```

Vamos implementar agora uma função que executa uma operação sobre todos os elementos das sublistas de uma lista de listas, usando funções de ordem superior.

```
map2 :: (a -> b) -> [[a]] -> [[b]]
map2 = undefined
```

E se pra cada lista eu quiser uma função diferente?

```
exemploFuncoes :: [(Int -> Int)]
exemploFuncoes = [(+ 1), (+ 2), (* 2)]
```

```
mapApply :: [(a -> b)] -> [[a]] -> [[b]]
mapApply = undefined
```

Pra finalizar, considere a seguinte definição sobre números inteiros.

```
data NatSemZero = One
                | Suc NatSemZero
                deriving (Eq, Ord, Show)

data Inteiro = Neg NatSemZero
            | Zero
            | Pos NatSemZero
            deriving (Eq, Show)
```

Vamos definir a seguinte função que converte um número inteiro nessa representação. Depois faremos o caminho inverso.

```
intToInteiro :: Int -> Inteiro
intToInteiro = undefined
```

```
inteiroToInt :: Inteiro -> Int
inteiroToInt = undefined
```