

Universidade Federal do ABC  
MCTA016-13 - Paradigmas de Programação  
2019.Q2

**Aula Prática 06 - Sistemas de voto**

Prof. Emílio Francesquini

25 de julho de 2019

Para esse exercício crie um novo projeto usando o `stack` com o nome `votos`. O conteúdo inicial de `Main.hs` deve ser:

```
module Main where

import Data.List

votos :: [String]
votos = ["Vermelho", "Azul", "Verde", "Azul", "Azul", "Vermelho"]

votosRanks :: [[String]]
votosRanks = ["Vermelho", "Verde"],
              ["Azul"],
              ["Verde", "Vermelho", "Azul"],
              ["Azul", "Verde", "Vermelho"],
              ["Verde"]

main :: IO ()
main = do
  print $ vencedor votos
  print $ vencedor' votosRanks
```

Na primeira parte do exercício vamos implementar a contagem de um sistema de votos simples, em que cada eleitor pode votar em apenas um candidato. O candidato vencedor é aquele com o maior número de votos. Para isso defina as seguintes funções:

```
-- conta quantos votos o candidato x recebeu
conta :: Eq a => a -> [a] -> Int
```

```

-- retorna a lista de elementos unicos
unicos :: Eq a => [a] -> [a]

-- retorna uma lista de pares ordenados (votos, candidato) com o total
-- de votos obtido por cada candidato use a função sort para ordenar
-- do menos para o mais votado
resultado :: Ord a => [a] -> [(Int,a)]

-- retorna o vencedor da eleição
vencedor :: Ord a => [a] -> a

```

O segundo sistema de voto é um pouco mais complexo. Cada eleitor pode votar em mais de um candidato na ordem de sua preferência. O processo de contagem é feito o seguinte procedimento:

- Elimina-se todos os votos vazios
- O candidato com o menor número de votos de primeira escolha é eliminado (*i.e.* remova todos seus votos da lista de votos)
- Repete-se até sobrar apenas um candidato

Para o exemplo acima, temos:

```

-- Inicial
[["Vermelho", "Verde"],
 ["Azul"],
 ["Verde", "Vermelho", "Azul"],
 ["Azul", "Verde", "Vermelho"],
 ["Verde"]]
-- Após a primeira rodada
[["Verde"],
 ["Azul"],
 ["Verde", "Azul"],
 ["Azul", "Verde"],
 ["Verde"]]
-- Após a segunda rodada
[["Verde"],
 [],
 ["Verde"],
 ["Verde"],
 ["Verde"]]

```

Após remover o segundo voto (vazio), nos sobra apenas um candidato ("Verde") que é declarado vencedor.

Para implementar esse sistema de voto, defina as seguintes funções:

```
-- elimina as listas vazias de uma lista de listas
rmvazio :: Eq a => [[a]] -> [[a]]

-- elimina um candidato da lista de votos
elimina :: Eq a => a -> [[a]] -> [[a]]

-- retorna uma lista dos candidatos existentes, do menos para o mais votado
rank :: Ord a => [[a]] -> [a]

-- retorna o vencedor executando o processo descrito acima
vencedor' :: Ord a => [[a]] -> a
```

Verifique pelo menos as seguintes propriedades com QuickCheck:

- O retorno da função `resultado` está realmente ordenado
- O resultado da função `rmvazio` não contém listas vazias