

# Linguagem Haskell

## 1. Objetivo

O objetivo deste estudo é praticar e entender os conceitos de programação funcional. As aplicações foram desenvolvidas usando a linguagem de programação Haskell (<https://haskell.org>).

## 2. Instruções

Para cada um dos problemas a seguir, foi desenvolvido um programa em Haskell que os resolva. Não foram utilizadas funções pré-definidas da biblioteca de Haskell ou qualquer outra de terceiros, sendo necessário, em alguns casos, reimplementar alguma função já existente.

- 1) Uma função recursiva para o cálculo de potência de dois números inteiros, onde o primeiro número é elevado ao segundo (Não utilizando o operador de potência (^)).

Ex.: **> potencia 2 3** --  $8=2*2*2$   
8

- 2) Calcular o somatório dos elementos ímpares de uma lista de inteiros.

Ex.: **> somalmpares [1,3,2,7,4,6,5]** --  $16=1+3+7+5$   
16

- 3) Substituir todos elementos de um determinado valor de uma lista de inteiros por um outro valor.

Ex.: **> substituir 1 0 [1,2,1,3,1]**  
[0,2,0,3,0]

- 4) Verificar se um número é primo.

Ex.: **> primo 17**  
True  
**> primo 0**  
False

- 5) Verifique se um número é perfeito, isto é, é igual a soma de seus divisores (exceto o próprio número).

Ex.: **> perfeito 28** --  $28=1+2+4+7+14$   
True

- 6) Função que retorna uma lista com a representação em binário de um número inteiro.

Ex.: **> binario 20**  
[1,0,1,0,0]

- 7) Verificar se todos os elementos de uma lista são distintos.

Ex.: **> distintos [1,2,4,2,5]**  
False  
**> distintos [3,2,1]**  
True

- 8) Verificar se duas listas são disjuntas.  
Ex.: > **disjuntas** [1,2,3] [5,4,6,0]  
True
- 9) Verificar se uma lista de inteiros é palíndromo.  
Ex.: > **palindromo** [1,2,3,4,3,2,1]  
True
- 10) Calcular todas as somas parciais de uma lista de inteiros.  
Ex.: > **somaParciais** [1,2,3,4] -- [1,1+2,1+2+3,1+2+3+4]  
[1, 3, 6, 10]
- 11) Linearizar uma lista de listas de inteiros.  
Ex.: > **linearizar** [ [1,2], [5], [0,4,2] ]  
[1,2,5,0,4,2]
- 12) Deslocar todos elementos de uma lista de inteiros k posições para a esquerda.  
Ex.: > **shift 3** [1,5,6,7,3,4,1] -- k=3  
[7,3,4,1,1,5,6]
- 13) Remover os n últimos elementos de uma lista de inteiros.  
Ex.: > **removerFim 2** [1,2,3,4,5,6] -- n=2  
[1,2,3,4]
- 14) Dadas duas listas ordenadas de forma crescente, obter a lista ordenada resultante da intercalação delas.  
Ex.: > **intercalar** [1,5,10] [2,7,9,20,25]  
[1,2,5,7,9,10,20,25]
- 15) Desenvolver uma solução para um quiosque de saque eletrônico que, para um determinado valor, deve entregar o menor número de cédulas de R\$1, R\$5, R\$10, R\$50 e R\$100, da menor para a maior.  
Ex.: > **trocar 162**  
[1, 1, 10, 50, 100]

## Observação

Cada programa está em um arquivo com o nome da função do item seguido de *.hs*. Por exemplo, para o primeiro item o arquivo foi nomeado *potencia.hs*, enquanto para o segundo *somaImpares.hs*, e assim por diante.