

## Prova de Haskell – Geradores de Listas

---

Nome: \_\_\_\_ Data: \_\_/\_\_/\_\_

**Conteúdo: Geradores de Listas, Compreensão de Listas, Recursão e Funções de Alta Ordem**

---

### Questões

#### 1. (Intermediária)

Explique a diferença entre as seguintes expressões e diga qual lista cada uma produz:

```
a = [x | x <- [1..10], even x]
b = [x*2 | x <- [1..10], x `mod` 3 == 0]
```

Reescreva ambas usando `filter` e `map`.

---

#### 2. (Intermediária)

Usando geradores de listas, crie uma função que gere todas as combinações possíveis de dois números entre 1 e 5, onde o primeiro é menor que o segundo.

---

#### 3. (Difícil)

Implemente uma função `pitagoricos :: Int -> [(Int, Int, Int)]` que gere todos os triplos pitagóricos `(a,b,c)` com `a^2 + b^2 == c^2` e `c <= n`.

---

#### 4. (Intermediária)

Explique a diferença entre:

```
lista1 = [(x,y) | x <- [1,2,3], y <- [4,5]]
lista2 = [(x,y) | y <- [4,5], x <- [1,2,3]]
```

---

#### 5. (Difícil)

Crie `primosAte :: Int -> [Int]` que gere todos os números primos até `n`.

---

## 6. (Intermediária)

Explique o funcionamento de:

```
pares = [(x,y) | x <- [1..4], y <- [x..4]]
```

---

## 7. (Difícil)

Crie `matriz :: Int -> Int -> [(Int, Int)]` que gere todas as posições (i,j) de uma matriz m×n.

---

## 8. (Intermediária/Difícil)

Escreva `intercalar :: [a] -> [a] -> [a]` que intercala dois vetores de mesmo tamanho usando `zip`.

---

## 9. (Difícil)

Defina `perfeitos :: Int -> [Int]` que gere todos os números perfeitos menores que n.

---

## 10. (Difícil)

Explique o que faz:

```
lista = [(x, y) | x <- [1..5], y <- [1..5], gcd x y == 1]
```

E crie `coprimosAte :: Int -> [(Int, Int)]` que retorne todos os pares coprimos até n.

---

**Fim da Prova**