

Gabarito – Prova de Haskell (Geradores de Listas)

1.

```
a = [2,4,6,8,10]
b = [6,12,18]
a = filter even [1..10]
b = map (*2) (filter (\x -> x `mod` 3 == 0) [1..10])
```

2.

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

3.

```
pitagoricos n = [(a,b,c) | c <- [1..n], b <- [1..c], a <- [1..b], a^2 + b^2
== c^2]
```

4.

- `lista1` = [(1,4),(1,5),(2,4),(2,5),(3,4),(3,5)]
- `lista2` = [(1,4),(2,4),(3,4),(1,5),(2,5),(3,5)]
- A ordem dos geradores altera a sequência dos resultados.

5.

```
primosAte n = [x | x <- [2..n], all (\d -> x `mod` d /= 0) [2..x-1]]
```

6.

- Gera pares com `y >= x`.
- Resultado: [(1,1),(1,2),(1,3),(1,4),(2,2),(2,3),(2,4),(3,3),(3,4),(4,4)]
- Equivalente a:

```
concatMap (\x -> map (\y -> (x,y)) [x..4]) [1..4]
```

7.

```
matriz m n = [(i,j) | i <- [1..m], j <- [1..n]]
```

8.

```
intercalar xs ys = concat [[x,y] | (x,y) <- zip xs ys]
```

9.

```
perfeitos n = [x | x <- [2..n], sum [d | d <- [1..x-1], x `mod` d == 0] == x]
```

10.

```
coprimosAte n = [(x,y) | x <- [1..n], y <- [1..n], gcd x y == 1]
```

- Retorna pares `(x,y)` entre 1 e `n` que são coprimos ($\text{gcd} == 1$).