INSTITUTO FEDERAL
Catarinense | Campus Rio do Sul

Curso: Bacharel em Ciência da Computação

Disciplina: Programação Lógica e Funcional

Semestre: 2023/1

Professor: Daniel Gomes Soares

Acadêmico: Rafael José Camargo Bekhauser

LISTA DE EXERCÍCIOS I

Código fonte disponível em: <https://github.com/rafandoo/Programacao-Logica-e-Funcional/tree/main/Funcional/ATV_1_LISTA_1>.

1.1

```
distanciaPontos :: (Float, Float) -> (Float, Float) -> Float
distanciaPontos (x1, y1) (x2, y2) = sqrt((x2-x1)^2 + (y2-y1)^2)
```

1.2

```
def distanciaPontos(x1, y1, x2, y2):
    return ((x2 - x1)**2 + (y2 - y1)**2)**0.5
```

2.1

```
teoremaPitagoras :: Float -> Float -> Float
teoremaPitagoras ca co = sqrt(ca^2 + co^2)
```

2.2

```
def teoremaPitagoras(ca, co):
    return (ca**2 + co**2)**(0.5)
```

3.1

```
areaRetangulo :: Float -> Float -> Float
areaRetangulo a b = a * b
```

3.2

```
def areaRetangulo(base, altura):
    area = base * altura
    return area
```

4.1

```
areaCirculo :: Float -> Float
areaCirculo r = pi * r * r
```

4.2

```
def areaCirculo(raio):
    return 3.14 * raio * raio
```

5.1

```
maiorMenor n = if n > 10 && n < 80 then True else False
```

5.2

```
def maiorMenor(num):
    if num > 10 and num < 80:
        return True
    return False
```

6.1

```
posneg n = if n < 0 then "negativo" else "positivo"
```

6.2

```
def posNeg(num):
    if num < 0:
        return 'negativo'
    return 'positivo'
```

7.1

```
parimpar n = if mod n 2 == 0 then "par" else "impar"
```

7.2

```
def parImpar(num):
    if num % 2 == 0:
        return 'par'
    return 'impar'
```

8.1

```
raizOuQuadrado :: Int -> Int
raizOuQuadrado n
    | even n    = floor $ sqrt $ fromIntegral n
    | otherwise = n ^ 2
```

8.2

```
def ehPar(num):
    if num % 2 == 0:
        return True
    return False

def calc(num):
    if ehPar(num):
        return num ** 0.5
    return num ** 2
```

9.1

```
categorias = ["INFANTIL A", "INFANTIL B", "JUVENIL A", "JUVENIL B",
"SENIOR"]

verificaCategoria idade
    | idade >= 5 && idade <= 7 = categorias !! 0
    | idade >= 8 && idade <= 10 = categorias !! 1
    | idade >= 11 && idade <= 13 = categorias !! 2
    | idade >= 14 && idade <= 17 = categorias !! 3
```

```
    | idade >= 18 = categorias !! 4
    | otherwise = "Categoria nao listada"
```

9.2

```
categorias = [
    "INFANTIL A",
    "INFANTIL B",
    "JUVENIL A",
    "JUVENIL B",
    "SENIOR"
]

def verificaCategoria(idade):
    if idade >= 5 and idade <=7:
        return("A categoria do competidor eh:
{}".format(categorias[0]))
    elif idade >= 8 and idade <= 10:
        return("A categoria do competidor eh:
{}".format(categorias[1]))
    elif idade >= 11 and idade <= 13:
        return("A categoria do competidor eh:
{}".format(categorias[2]))
    elif idade >= 14 and idade <= 17:
        return("A categoria do competidor eh:
{}".format(categorias[3]))
    elif idade >= 18:
        return("A categoria do competidor eh:
{}".format(categorias[4]))
    return("Categoria não listada")
```

10.1

```
mediaAluno n1 n2
    | media >= 7 = "Aprovado, nota: " ++ show media
    | media >= 3 = "Exame, nota: " ++ show media
    | otherwise = "Reprovado, nota: " ++ show media
    where media = (n1 + n2) / 2
```

10.2

```
def mediaAluno(n1, n2):
```

```
    media = (n1 + n2) / 2
    if media >= 7:
        return "Aprovado, nota: {}".format(media)
    elif media >= 3:
        return "Exame, nota: {}".format(media)
    return "Reprovado, nota: {}".format(media)
```

11.1

```
verificaEleitor idade
    | idade < 16 = "Nao eleitor"
    | idade >= 18 && idade <= 65 = "Eleitor obrigatorio"
    | otherwise = "Eleitor facultativo"
```

11.2

```
def verificaEleitor(idade):
    if idade < 16:
        return("Não eleitor")
    elif idade >= 18 and idade <= 65:
        return("Eleitor obrigatorio")
    return("Eleitor facultativo")
```

12.1

```
calculoLucro valor
    | valor < 30.00 = "O valor de venda sera R$" ++ show(valor +
(valor * 0.45))
    | otherwise = "O valor de venda sera R$" ++ show(valor + (valor
* 0.30))
```

12.2

```
def calculoLucro(valor):
    if valor < 30.00:
        return("O valor de venda sera R${:.2f}".format(valor +
(valor * 0.45)))
    return("O valor de venda sera R${:.2f}".format(valor + (valor *
0.30)))
```

13.1

```
ehBissexto ano = (ano `mod` 4 == 0 && ano `mod` 100 /= 0) || (ano
```

```
`mod` 400 == 0)
```

13.2

```
def ehBissexto(ano):
    if (ano % 4 == 0 and ano % 100 != 0) or ano % 400 == 0:
        return True
    return False
```

14.1

```
fatorial n = product [1..n]
```

14.2

```
def fatorial(n):
    if n == 0:
        return 1
    return n * fatorial(n-1)
```

15.1

```
ehPalindromo :: String -> Bool
ehPalindromo [] = True
ehPalindromo [x] = True
ehPalindromo (x:xs) = if x == last xs then ehPalindromo (init xs)
else False
```

15.2

```
def ehPalindromo(palavra):
    if palavra == palavra[::-1]:
        return True
    return False
```