

# FUNÇÕES: EXCEÇÕES, MODULARIZAÇÃO E MAIS. SOLUÇÕES AOS EXERCÍCIOS

Santiago Valdés Ravelo  
[https://ic.unicamp.br/~santiago/  
ravelo@unicamp.br](https://ic.unicamp.br/~santiago/ravelo@unicamp.br)

MC102 - Algoritmos e  
Programação de  
Computadores

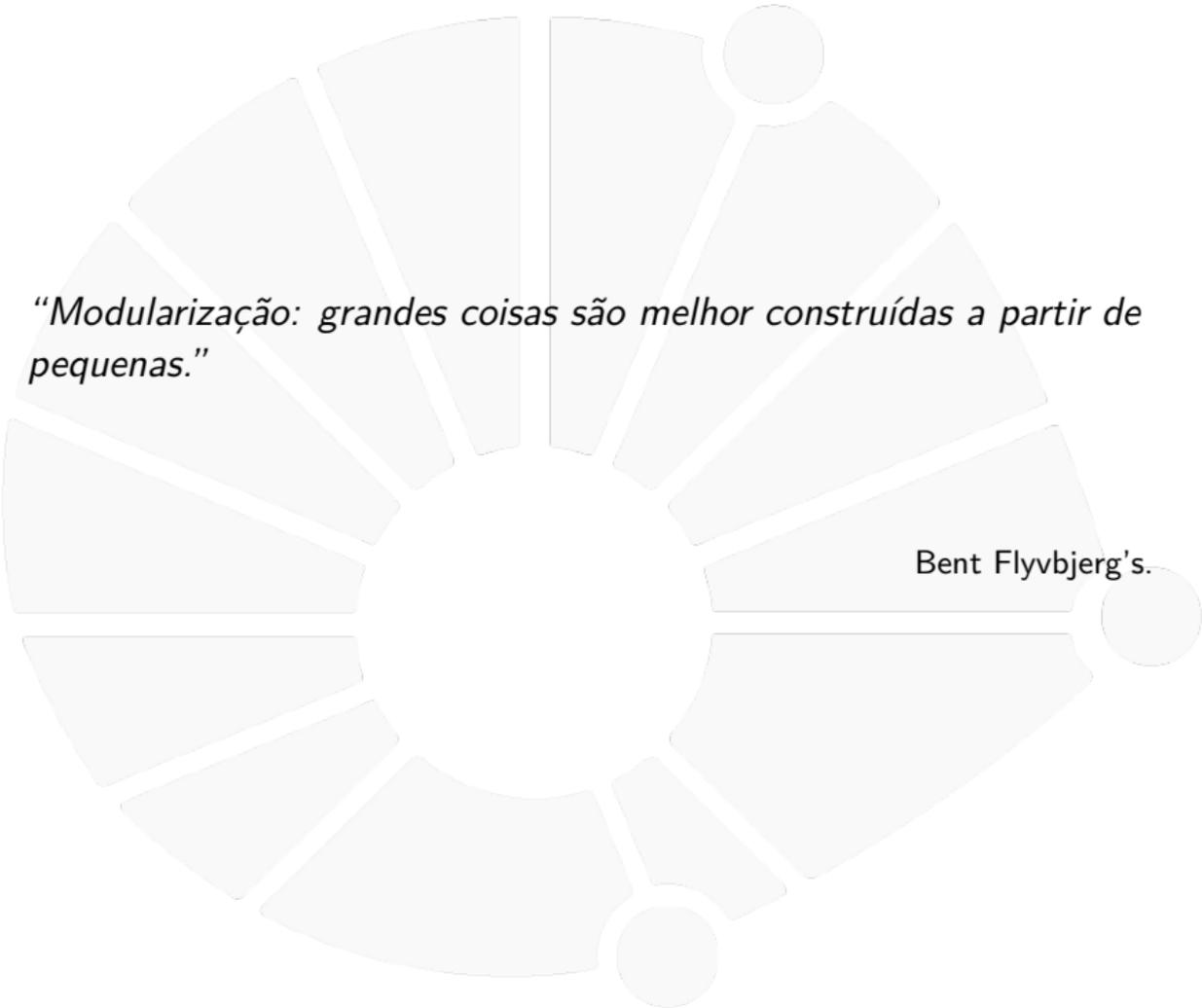
04/25

11



UNICAMP



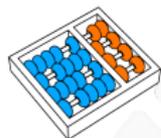


*“Modularização: grandes coisas são melhor construídas a partir de pequenas.”*

Bent Flyvbjerg's.



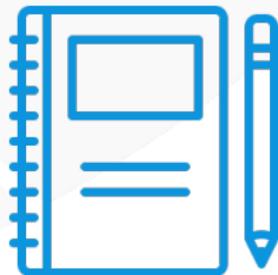
# EXERCÍCIOS

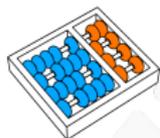


## Funções



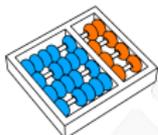
**Soluções para os exercícios!**





## Exercícios

1. Use a função **seleciona** e a biblioteca **math** para criar uma função que dada uma lista de números reais, devolve uma lista dos números **x** tal que **x** em radianos está no primeiro quadrante.
2. Faça uma função **mapeia** que, dada uma lista **l** e uma função de um parâmetro **f**, devolve uma nova lista onde **f** foi aplicada a cada elemento de **l**.
  - ▶ Por exemplo, se **l = [1, 2, 3]** e **f(x) = x \* x**, então a nova lista é **[1, 4, 9]**.
3. Use a função anterior para, dada uma lista, encontrar uma nova lista com todos os seus elementos elevado ao quadrado.
4. Faça uma função **combina** que permite aplicar uma função **f** sobre uma lista **l** para combinar os seus resultados e obter um único valor.
  - ▶ Ex: Podemos somar todos os elementos de uma lista de números.
  - ▶ Ex: Podemos multiplicar todos os elementos de uma lista de números.
  - ▶ Ex: Podemos fazer **and** de vários valores booleanos.
5. Use a função anterior para concatenar a representação decimal de uma lista de números inteiros positivos.
  - ▶ Ex: se a lista é **[1, 2, 0, 15]**, o resultado é **'12015'**.



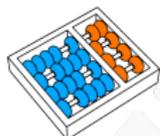
## Soluções

Seleciona primeiro quadrante (valores entre 0 e  $\frac{\pi}{2}$ ):

```
1 from math import pi
2
3 def seleciona(lista, funcao):
4     nova_lista = []
5     for x in lista:
6         if funcao(x):
7             nova_lista.append(x)
8     return nova_lista
9
10 def primeiro_quadrante(lista):
11     return seleciona(lista, lambda x: x >= 0 and 2 * x <= pi)
```

Mapeia e mapeia ao quadrado:

```
1 def mapeia(lista, funcao):
2     nova_lista = []
3     for x in lista:
4         nova_lista.append(funcao(x))
5     return nova_lista
6
7 def quadrados(lista):
8     return mapeia(lista, lambda x: x * x)
```



## Soluções. Continuação

Combina e usos: somando, multiplicando, fazendo and, concatenado:

```
1 def combina(lista, funcao):
2     resultado = None
3     for x in lista:
4         if resultado is None:
5             resultado = funcao(x)
6         else:
7             resultado = funcao(resultado, x)
8     return resultado
9
10 def soma(lista):
11     return combina(lista, lambda x, y = 0: x + y)
12
13 def multiplica(lista):
14     return combina(lista, lambda x, y = 1: x * y)
15
16 def faz_and(lista):
17     return combina(lista, lambda x, y = True: x and y)
18
19 def concatena(lista):
20     return combina(lista, lambda x, y = '': str(x) + str(y))
```

# FUNÇÕES: EXCEÇÕES, MODULARIZAÇÃO E MAIS. SOLUÇÕES AOS EXERCÍCIOS

Santiago Valdés Ravelo  
[https://ic.unicamp.br/~santiago/  
ravelo@unicamp.br](https://ic.unicamp.br/~santiago/ravelo@unicamp.br)

MC102 - Algoritmos e  
Programação de  
Computadores

04/25

11



UNICAMP

