

## Aula 6

### Linguagem de Programação

Prof. Wellington Rodrigo Monteiro

1

### Conversa Inicial

2

- Fugindo do Go Horse/Extreme Go Horse
- Pensando sobre usos futuros do software
  - Otimização
  - Concorrência
  - Escalabilidade
  - Protocolos/interfaces
- CPython

3

### Otimização

4

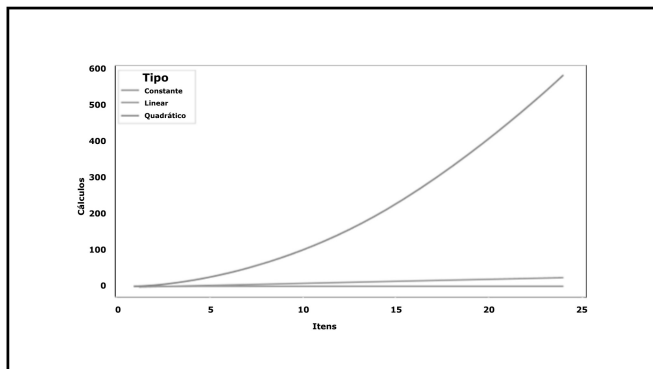
- Otimização de processos de desenvolvimento de software
- Otimização de algoritmos: eficiência

5

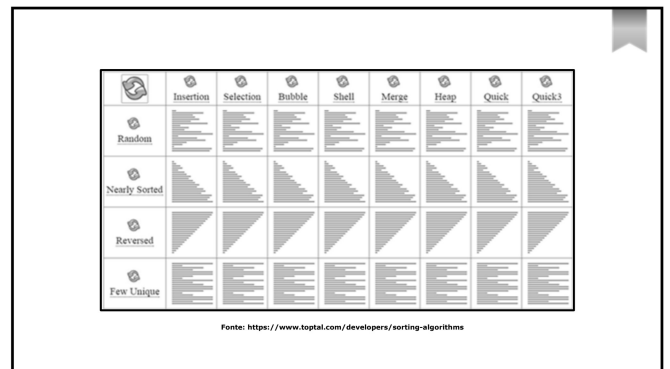
### Funções de complexidades distintas

```
1 # retorna a metade do último número
2 def operacao_constante(items):
3     contador = 1
4     return contador, items[-1]/2
5
6 # multiplica cada número da lista por dois
7 def operacao_linear(items):
8     contador = 0
9     novos_items = []
10    for item in items:
11        novos_items.append(item * 2)
12        contador += 1
13    return contador, novos_items
14
15 # multiplica cada número da lista por todos os números da mesma lista
16 def operacao_quadratica(items):
17     contador = 0
18     novos_items = []
19     for item in items:
20         for item_2 in items:
21             novos_items.append(item * item_2)
22             contador += 1
23     return contador, novos_items
```

6



7



8

**Concorrência**

9

- Processos
- Threads
- Processamento síncrono x assíncrono

10

**Escalabilidade**

11

- "Continua funcionando se houver um pico de acessos/uso?"
- Paralelização
  - Mais computadores para o mesmo trabalho
  - Mais memória/processador em uma mesma máquina

12

## Escalabilidade em Python

- Uso de generators
- Uso de iterators
- Uso de threading/processamento assíncrono
- Aplicações serverless (funções)

13

## CPython

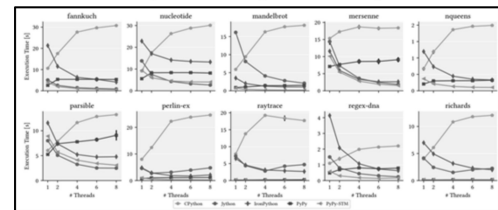
14

## Implementações de linguagens

- Linguagem
  - Python
  - C#
  - Java
- Implementação
  - Python: CPython, PyPy, IronPython, Jython
  - C#: .NET, Mono, DotGNU
  - Java: Oracle, OpenJDK

15

## Comparação de performance das implementações em Python



16

## Protocolos e Interfaces

17

- Interfaces
  - Definem apenas o padrão/quais métodos as classes devem ter
  - Não implementamos a lógica nos métodos
  - Python: módulo Protocol
- Classes abstratas
  - Podem definir os métodos e, opcionalmente, implementá-los
  - Python: módulo ABC
- Interpretação simplista (ex.: interface informal x formal)

18

### Exemplos

- **Classe abstrata: Animal** (faz som; se movimenta)
- **Interface: AnimalEstimacao** (passeia)
- **Classe: Cachorro** (faz som como um cachorro; se movimenta como um cachorro; passeia)

19

- **Classe abstrata: Veículo** (movimenta, carrega, descarrega)
- **Interface: Automóvel** (liga o motor, desliga o motor)
- **Classe: Sedan** (movimenta, carrega, descarrega, liga o motor, desliga o motor)

20