Bruno Felipe de Souza Araujo

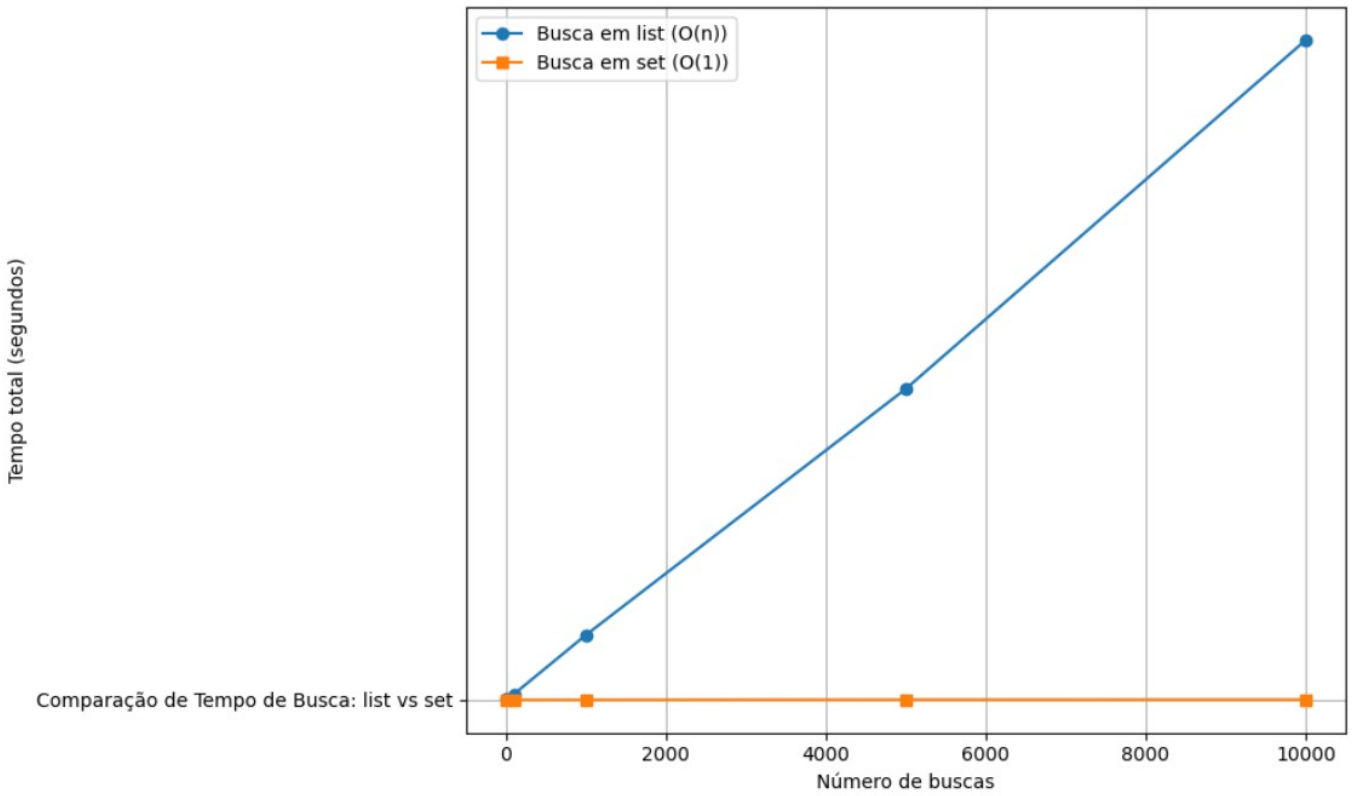
CDN 3 Semestre

```python
import random
import timeit

lista_pequena = random.sample(range(1,100), 10)
lista_grande = random.sample(range(1,20000), 10000)

alvo_pequeno = lista_pequena[-1]
alvo_grande = lista_grande[1]

codigo_teste_pequena = 'alvo_pequeno in lista_pequena'
codigo_teste_grande = 'alvo_grande in lista_grande'

tempo_pequena = timeit.timeit(stmt=codigo_teste_pequena, globals=globals(), number=1000)
tempo_grande = timeit.timeit(stmt=codigo_teste_grande, globals=globals(), number=1000)

print("🔍 Comparação de busca linear (último elemento)")
print(f"Lista com 10 elementos: {tempo_pequena:.6f} s (média: {tempo_pequena/1000:.10f} s)")
print(f"Lista com 10000 elementos: {tempo_grande:.6f} s (média: {tempo_grande/1000:.10f} s)")
```

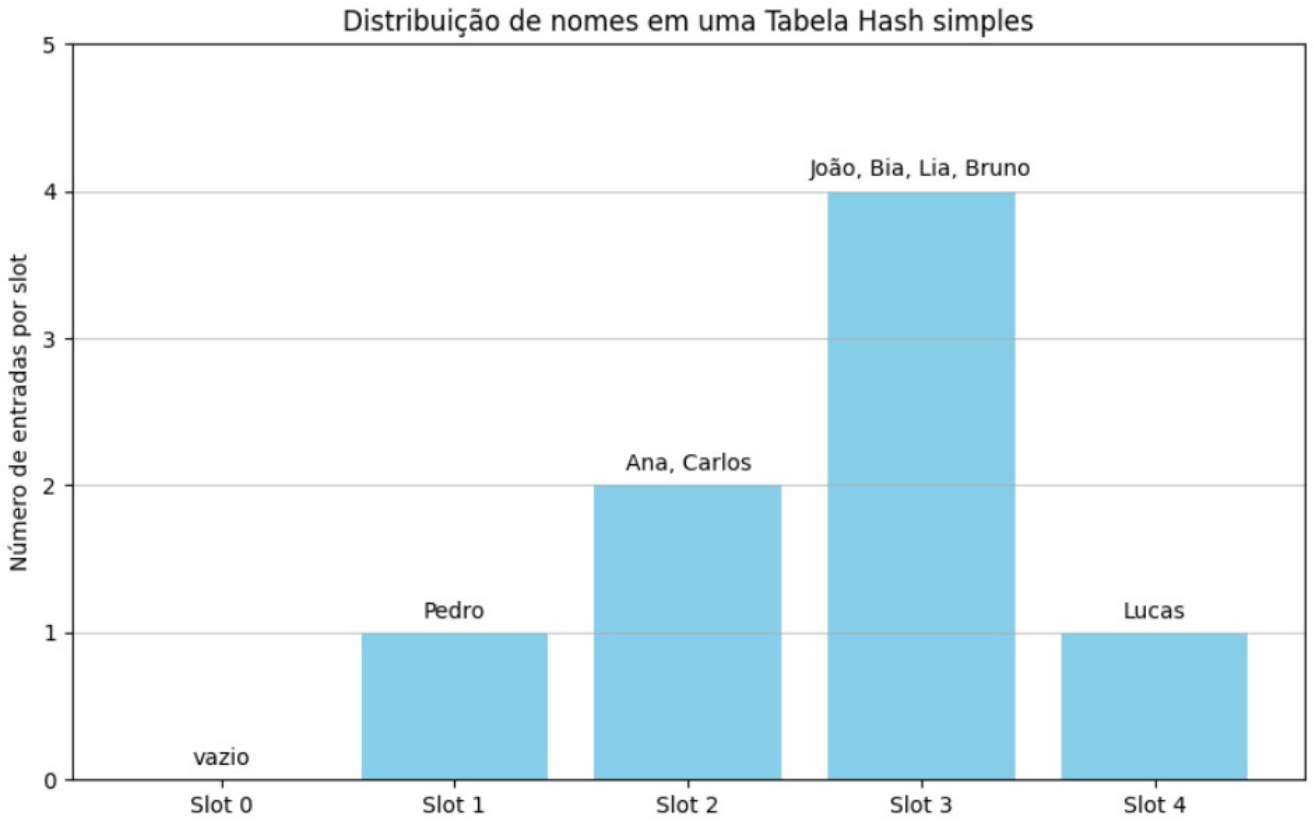
```
🔍 Comparação de busca linear (último elemento)
Lista com 10 elementos: 0.000622 s (média: 0.0000006221 s)
Lista com 10000 elementos: 0.000200 s (média: 0.0000001997 s)
```

```python
import random
import timeit
import matplotlib.pyplot as plt

dados = random.sample(range(1,20000), 10000)
alvo = dados[-1]
lista = dados
conjunto = set(dados)

quantidades = [10, 100, 1000, 5000, 10000]
tempos_lista = []
tempos_set = []

for n in quantidades:
    tempo_lista = timeit.timeit(stmt='alvo in lista', globals=globals(),  number=n)
    tempo_set = timeit.timeit(stmt='alvo in conjunto', globals=globals(), number=n)
    tempos_lista.append(tempo_lista)
    tempos_set.append(tempo_set)

print("🔍 Comparação de tempo total para diferentes quantidades de buscas:")
for i, n in enumerate(quantidades):
    print(f"{n:>5} buscas - list: {tempos_lista[i]:.6f}s | set: {tempos_set[i]:.6f}s")

plt.figure(figsize=(10,6))
plt.plot(quantidades, tempos_lista, marker='o', label='Busca em list (O(n))')
plt.plot(quantidades, tempos_set, marker='s', label='Busca em set (O(1))')
plt.plot('Comparação de Tempo de Busca: list vs set')
plt.xlabel('Número de buscas')
plt.ylabel('Tempo total (segundos)')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```

```
🔍 Comparação de tempo total para diferentes quantidades de buscas:
   10 buscas - list: 0.002151s | set: 0.000003s
  100 buscas - list: 0.021824s | set: 0.000007s
 1000 buscas - list: 0.248447s | set: 0.000050s
 5000 buscas - list: 1.191227s | set: 0.000884s
10000 buscas - list: 2.527246s | set: 0.000977s
```




```python
import matplotlib.pyplot as plt

tamanho = 5
entradas = ["João", "Ana", "Bia", "Carlos", "Lucas", "Lia", "Pedro", "Bruno"]
hash_slots = [[] for _ in range(tamanho)]

Tabnine | Edit | Test | Explain | Document
def hash_simples (chave):
    return sum(ord(c) for c in chave) % tamanho

for nome in entradas:
    indice = hash_simples(nome)
    hash_slots [indice].append(nome)

labels = [f"Slot {i}" for i in range(tamanho)]
valores = [len(slot) for slot in hash_slots]
conteudo = [", ".join(slot) if slot else "vazio" for slot in hash_slots]

fig, ax = plt.subplots(figsize=(10,6))
bars = ax.bar(labels, valores, color='skyblue')

for bar, texto in zip(bars, conteudo):
    yval = bar.get_height()
    ax.text(bar.get_x() + bar.get_width()/2, yval + 0.1, texto, ha='center', fontsize=10)

plt.title("Distribuição de nomes em uma Tabela Hash simples")
plt.ylabel("Número de entradas por slot")
plt.ylim(0, max (valores)+1)
plt.grid(axis='y', linestyle='-', alpha=0.7)
plt.show()
```




```python
from faker import Faker

fake = Faker('pt_BR')
clientes = {}

for _ in range(5):
    nome = fake.name()
    dados = {
        'CPF': fake.cpf(),
        'Email': fake.email(),
        'Endereço': fake.address(),
        'Data de Nascimento': fake.date_of_birth (minimum_age=18, maximum_age=65)
    }
    clientes [nome] = dados

for nome, info in clientes.items():
    print(f"Cliente: {nome}")
    for chave, valor in info.items():
        print(f" {chave}: {valor}")
print("-" * 40)
```

```
Cliente: Antonella Cavalcanti
 CPF: 587.649.102-02
 Email: almeidaravi-lucca@example.com
 Endereço: Trevo Pastor
Luxemburgo
48051428 Costa Alegre / RO
 Data de Nascimento: 1995-07-14
Cliente: Manuella Cunha
 CPF: 235.974.680-47
 Email: juliada-mota@example.com
 Endereço: Recanto Nunes, 44
Vila São Dimas
66283872 Vieira das Pedras / MS
 Data de Nascimento: 1981-04-08
Cliente: Sr. João Gabriel Barbosa
 CPF: 875.029.634-56
 Email: da-rosaotavio@example.com
 Endereço: Campo Maria Luísa Pinto, 22
Nova Suíça
04308-770 da Paz / RS
 Data de Nascimento: 1996-02-22
Cliente: Sr. Levi Novaes
 CPF: 825.971.436-19
 Email: eda-costa@example.net
 Endereço: Recanto da Rocha, 789
Santa Terezinha
97149748 da Rosa de Andrade / ES
 Data de Nascimento: 1981-01-24
Cliente: Pietro Sousa
 CPF: 218.435.760-07
 Email: leaoalice@example.com
 Endereço: Aeroporto Oliveira, 22
Vila Califórnia
54898-357 Viana de Goiás / PE
 Data de Nascimento: 1983-07-10
```

```python
from faker import Faker
import timeit
import random

fake = Faker()
dict_10 = {fake.name(): fake.email() for _ in range(10)}
chave_10 = list(dict_10.keys()) [-1]
dict_10000 = {fake.name(): fake.email() for _ in range(10000)}
chave_10000 = list(dict_10000.keys())[-1]
tempo_10 = timeit.timeit(stmt="chave_10 in dict_10", globals=globals(), number=10000)
tempo_10000 = timeit.timeit(stmt="chave_10000 in dict_10000", globals=globals(), number=10000)
print(" Tempo total para 10.000 buscas:")
print(f"Dicionário com     10 elementos: {tempo_10:.6f} segundos")
print(f"Dicionário com 10.000 elementos: {tempo_10000:.6f} segundos")
```

```
 Tempo total para 10.000 buscas:
Dicionário com     10 elementos: 0.000590 segundos
Dicionário com 10.000 elementos: 0.000853 segundos
```

```python
coordenadas = (3.5, 7.2, 0.0)
print("Tupla:", coordenadas)

try:
    coordenadas[0] = 10
except TypeError as e:
    print("Erro ao tentar alterar a tupla:", e)

print("Segundo valor (índice 1):", coordenadas [1])
print("Existe o valor 7.2?", 7.2 in coordenadas)
PI = (3.14159,)
RGB_BRANCO = (255, 255, 255)
dias_da_semana = ('segunda', 'terça', 'quarta', 'quinta', 'sexta', 'sábado', 'domingo')
cidades = {
    ('São Paulo', 'SP'): 12_000_000,
    ('Rio de Janeiro', 'RJ'): 6_700_000
}
print("População de São Paulo:", cidades[('São Paulo', 'SP')])
```

```
Tupla: (3.5, 7.2, 0.0)
Erro ao tentar alterar a tupla: 'tuple' object does not support item assignment
Segundo valor (índice 1): 7.2
Existe o valor 7.2? True
População de São Paulo: 12000000
```