



# Vetor



**INSTITUTO  
FEDERAL**

Paraíba

---

Campus  
João Pessoa

## ≡ Problema

- Em diversas situações é necessário armazenar informações para que seja possível realizar alguma atividade específica.

### **Por exemplo:**

- Obter o nome de 30 pessoas, classificar e exibir em ordem alfabética;
- Obter o preço de 1000 produtos e exibir os mais caros considerando empate;
- Armazenar os números (50) de um aposta da lotomania para que se possa verificar dentre 50 apostadores quais os premiados.

## ≡ Cenário para Análise

- Vamos partir de um programa, em Python, para ler N números inteiros, calcular e exibir a média dos números lidos.
- Seria algo do tipo ...

```
n = 5
soma = 0
print(f'Digite os {n} números:')
for i in range(n):
    numero = int(input())
    soma = soma + numero
media = soma / n
print(f'Média = {media}')
```

Digite os 5 números:

8

4

6

2

7

Média = 5.4

## ≡ Ainda no mesmo cenário ...

E se eu quiser saber quais dos **N números lidos** estavam acima da média?!

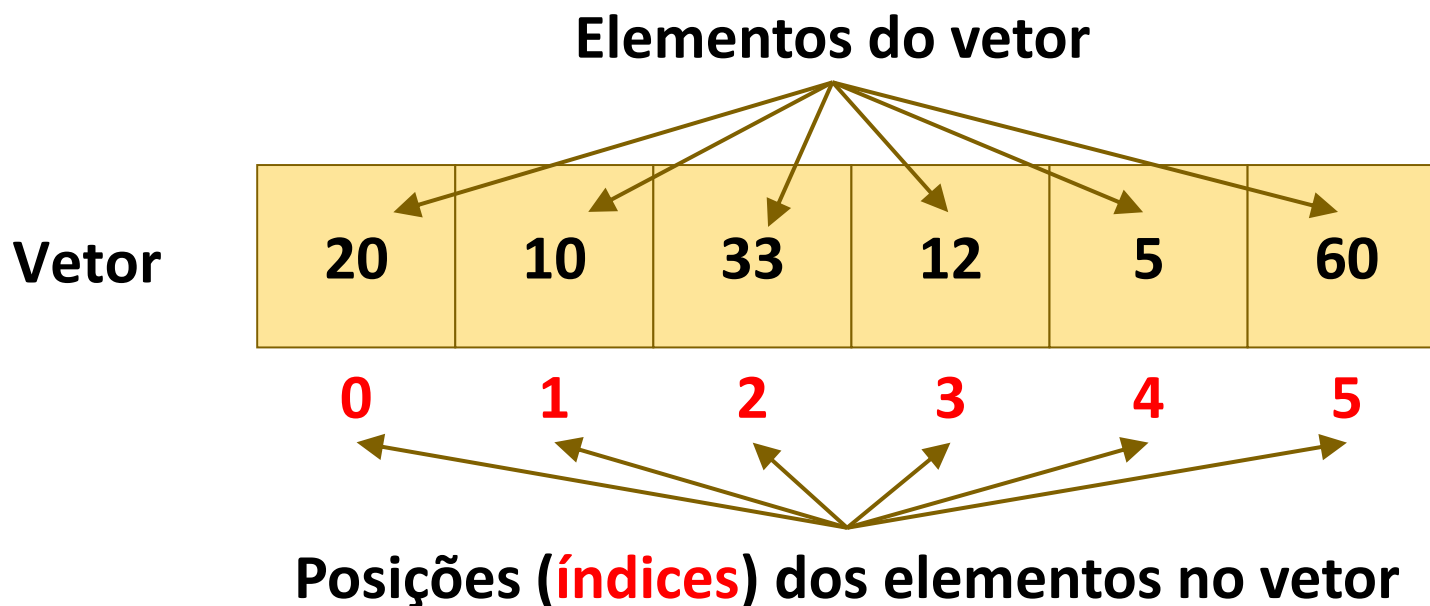
E se esse **N** for igual a **100**?



**Preciso declarar 100 variáveis?!?!?!?**

## ≡ Solução!

- É possível definir variáveis que guardam mais de um valor de um mesmo tipo.
- Essas variáveis são conhecidas como arranjo (*array*) ou vetor.
- Graficamente, representamos assim um **vettor**:

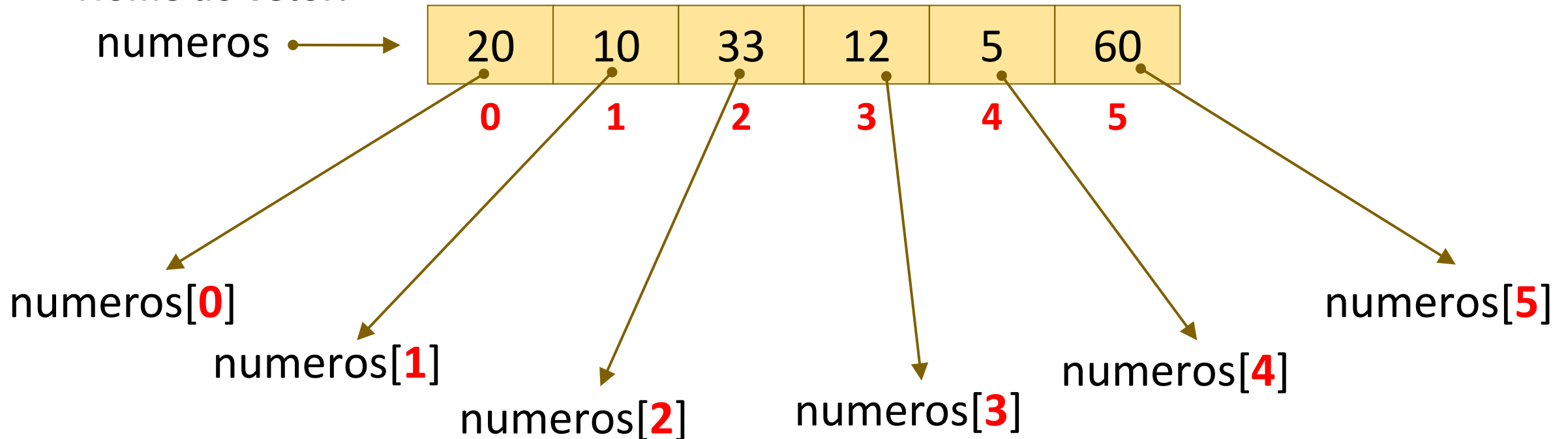




# Vetor

- O **vetor** possui um **identificador** e cada elemento que o compõe pode ser **individualizado/referenciado por meio de um índice**

Nome do vetor:



## ≡ Em Python...

- Em Python existem módulos que nos permitem manipular de forma eficiente vetores. Entretanto, nesse curso, **queremos aprender a representar um vetor** e a **definir operações** de manipulação sobre ele.
- Em Python existem três tipos principais de variáveis compostas:
  - Listas
  - Tuplas
  - Dicionários
- Para implementar um vetor em Python utilizaremos uma variável do tipo lista (***list***), porém não usaremos o conceito e os métodos deste tipo.
- Em Python os elementos de uma lista podem ser de qualquer tipo, porém em muitas linguagens os elementos precisam ser do mesmo tipo.



## Inicializando um vetor

- Inicializando um vetor com valores predefinidos:

```
nome = ['João', 'Maria', 'Pedro']  
print(nome)  
print(nome[0])  
print(nome[1])  
print(nome[2])
```

```
['João', 'Maria', 'Pedro']  
João  
Maria  
Pedro
```

- Inicializando um vetor com valores nulos:

```
vetor = [None]*5  
print(vetor)
```

```
[None, None, None, None, None]
```



## ≡ Utilizando *for* para percorrer um vetor

- Para acessarmos todos os elementos de um vetor, podemos utilizar o comando *for*.
- Exemplo: Lendo 3 nomes e armazenando em um vetor

```
n = 3
v = [None]*n
for i in range(n):
    v[i] = input()
print(v)
```

```
João
Maria
Pedro
['João', 'Maria', 'Pedro']
```

## ≡ Utilizando *for* para percorrer um vetor

- Exemplo: Exibindo os elementos de um vetor, sendo um por linha.

```
nome = ['João', 'Maria', 'Pedro']  
for i in range(3):  
    print(nome[i])
```

```
João  
Maria  
Pedro
```

- Utilizando o próprio vetor como objeto iterável do *for*:

```
nome = ['João', 'Maria', 'Pedro']  
for e in nome:  
    print(e)
```

```
João  
Maria  
Pedro
```

## ≡ Cuidados ao manipular vetores

- Acessar uma posição inválida de um vetor causará um erro na execução do programa.
- Exemplo:

```
nome = ['João', 'Maria', 'Pedro']  
print(nome[3])
```

```
-----  
IndexError                                Traceback (most recent call last)  
<ipython-input-10-1f99d90a7be7> in <module>()  
      1 nome = ['João', 'Maria', 'Pedro']  
----> 2 print(nome[3])
```

**IndexError:** list index out of range

SEARCH STACK OVERFLOW

## ≡ Cuidados ao manipular vetores

- Um vetor não pode ser atribuído diretamente para outro vetor.
- Ex: Suponha que eu queira copiar os elementos de um vetor *a* para um vetor *b*.

```
a = [1,2,3]
b = a
print('a =',a)
print('b =',b)
```

```
a[1] = 0
print()
print('a =',a)
print('b =',b)
```

```
a = [1, 2, 3]
b = [1, 2, 3]
```

```
a = [1, 0, 3]
b = [1, 0, 3]
```

**ERRADO**

- Note que *a* e *b* tornaram-se o mesmo vetor, apenas com nomes diferentes.

## ≡ Cuidados ao manipular vetores

- Solução para resolver o problema anterior (copiar os elementos de um vetor para outro):

**CORRETO**

```
n = 3
a = [1,2,3]
b = [None]*n
for i in range(n):
    b[i] = a[i]
print('a =',a)
print('b =',b)

a[1] = 0
print()
print('a =',a)
print('b =',b)
```

```
a = [1, 2, 3]
b = [1, 2, 3]

a = [1, 0, 3]
b = [1, 2, 3]
```

## ≡ Resolvendo o problema do cenário inicial

- Ler N números inteiros, calcular e exibir a média dos números lidos e mostrar quais os números acima dessa média.

```
n = 5
vetor = [None]*n
soma = 0
print(f'Digite os {n} números:')
for i in range(n):
    vetor[i] = int(input())
    soma = soma + vetor[i]
media = soma / n
print(f'Média = {media}')
print(f'Os números acima da média são:')
for e in vetor:
    if e > media:
        print(e)
```

```
Digite os 5 números:
8
4
6
2
7
Média = 5.4
Os números acima da média são:
8
6
7
```