

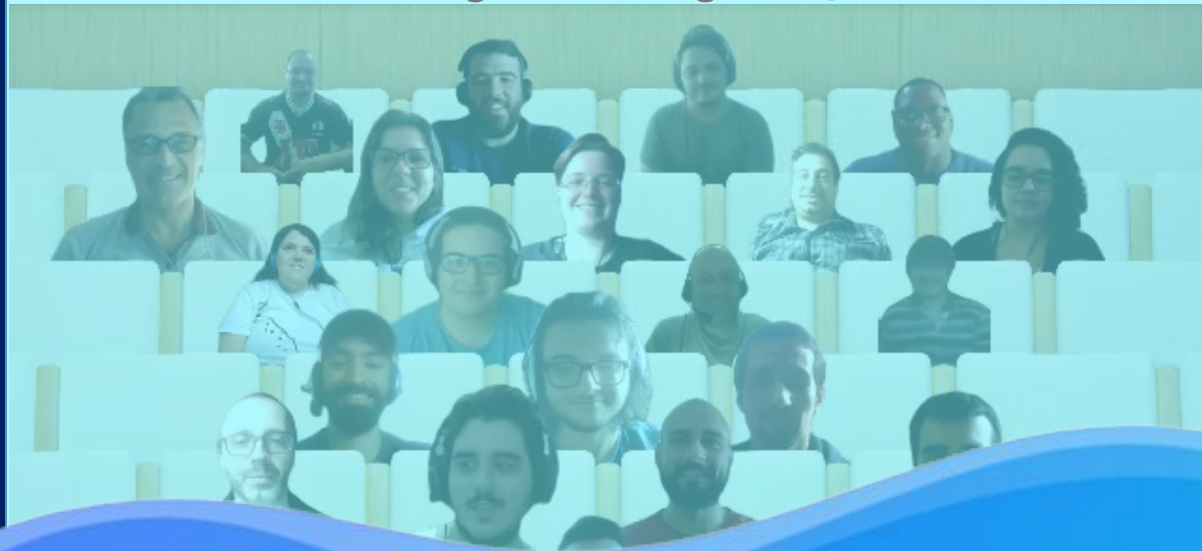
53 45 52 45 49 20 46 49 45 4c 20  
41 4f 53 20 50 52 45 43 45 49 54  
4f 53 20 44 41 20 48 4f 4e 52 41  
20 45 20 44 41 20 43 49 c3 8a 4e  
43 49 41 2c 20 50 52 4f 4d 4f 56  
45 4e 44 4f 20 4f 20 55 53 4f 20  
45 20 4f 20 44 45 53 45 4e 56 4f  
4c 56 49 4d 45 4e 54 4f 20 44 41  
20 49 4e 46 4f 52 4d c3 81 54 49  
43 41 20 45 4d 20 42 45 4e 45 46  
c3 8d 43 49 4f 20 44 4f 20 43 49  
44 41 44 c3 83 4f 20 45 20 44 41  
20 53 4f 43 49 45 44 41 44 45 2e

## RESIDÊNCIA DE SOFTWARE

**CAPACITAR**  
**TREINAR**  
**EMPREGAR**  
**TRANSFORMAR**



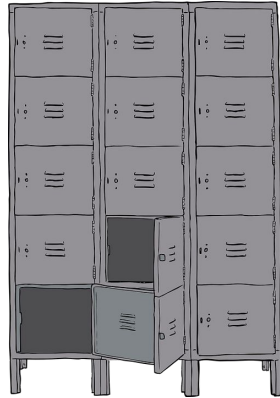
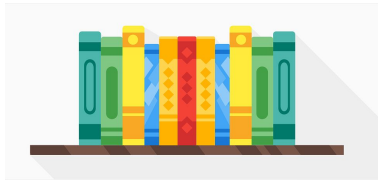
## Lógica de Programação



Aula 4 - Vetores  
Roni Schanuel  
18-03-2022

# Estrutura de Dados

“Estrutura de dados é o ramo da computação que estuda os diversos mecanismos de organização de dados para atender aos diferentes requisitos de processamento.” – RICARTE, IVAN LUIZ MARQUES ( UNICAMP )



# Estruturas de dados : Conceitos

Uma estrutura de dados pode ser dividida em dois pilares fundamentais : **dado** e **estrutura**.

## DADO

Dados são qualquer sequência de um ou mais símbolos que tenham significado por ato(s) específico(s) de interpretação.



## ESTRUTURA

Elemento estrutural responsável por carregar as informações dentro de uma estrutura de software

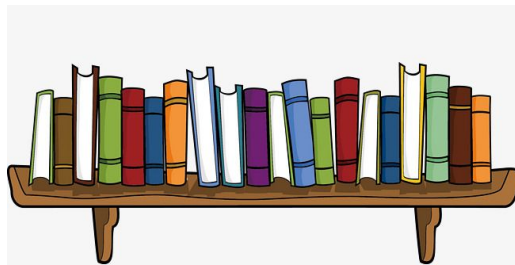
# Estruturas de dados : Conceitos

Uma estrutura de dados pode ser dividida em dois pilares fundamentais : **dado** e **estrutura**.

## DADO

Tipos de dados :

- Inteiro
- Ponto flutuante
- Caractere
- Texto



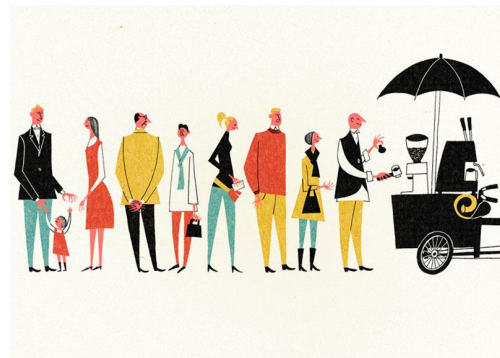
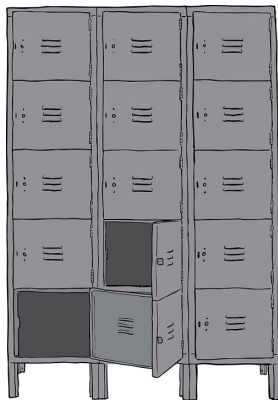
## ESTRUTURA

Tipos de estruturas:

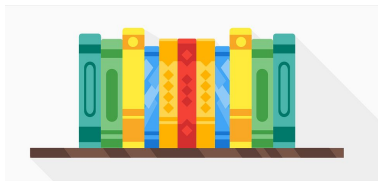
- Vetores
- Pilhas
- Filas
- Listas

# Principais tipos de estruturas de dados

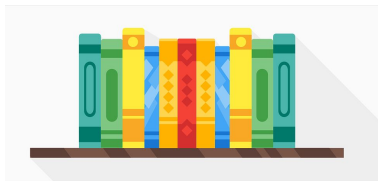
- Vetores
  - Unidimensionais
  - Bidimensionais ( Matrizes )
- Pilhas (não estudaremos agora)
- Filas (não estudaremos agora)



# Vetores

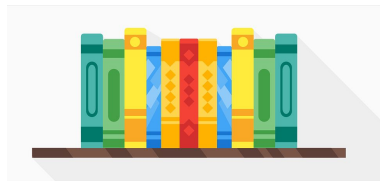


# Vetores



| Tipo | Nome | Capacidade |
|------|------|------------|
|      |      |            |

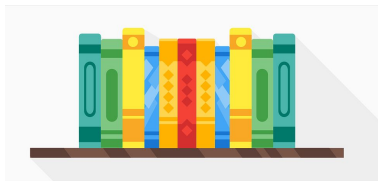
# Vetores



| Tipo   | Nome | Capacidade |
|--------|------|------------|
| livros |      |            |

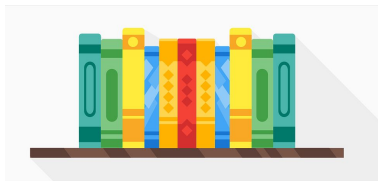


# Vetores



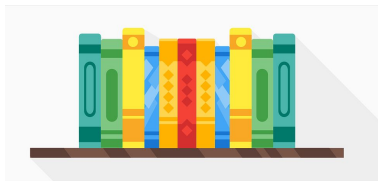
| Tipo   | Nome         | Capacidade |
|--------|--------------|------------|
| livros | minhaEstante |            |

# Vetores



| Tipo   | Nome         | Capacidade |
|--------|--------------|------------|
| livros | minhaEstante | 11         |

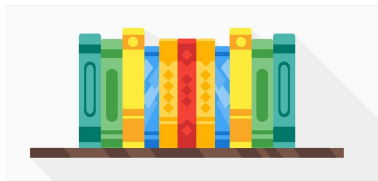
# Vetores



| Tipo   | Nome         | Capacidade |
|--------|--------------|------------|
| livros | minhaEstante | 11         |

livros minhaEstante[11];

# Vetores



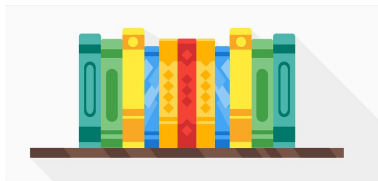
| Tipo   | Nome         | Capacidade |
|--------|--------------|------------|
| livros | minhaEstante | 11         |

livros minhaEstante[11]

|   |    |    |     |   |
|---|----|----|-----|---|
| 1 | 26 | 22 | 100 | 2 |
|---|----|----|-----|---|

| Tipo | Nome | Capacidade |
|------|------|------------|
|      |      |            |

# Vetores



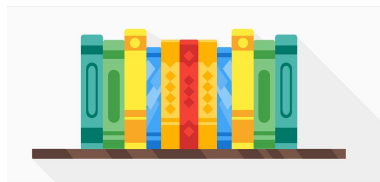
| Tipo   | Nome         | Capacidade |
|--------|--------------|------------|
| livros | minhaEstante | 11         |

livros minhaEstante[11]

|   |    |    |     |   |
|---|----|----|-----|---|
| 1 | 26 | 22 | 100 | 2 |
|---|----|----|-----|---|

| Tipo    | Nome | Capacidade |
|---------|------|------------|
| inteiro |      |            |

# Vetores



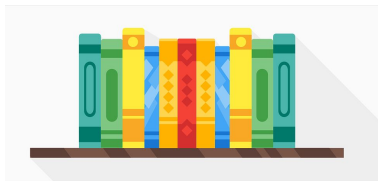
| Tipo   | Nome         | Capacidade |
|--------|--------------|------------|
| livros | minhaEstante | 11         |

livros minhaEstante[11]

|   |    |    |     |   |
|---|----|----|-----|---|
| 1 | 26 | 22 | 100 | 2 |
|---|----|----|-----|---|

| Tipo    | Nome     | Capacidade |
|---------|----------|------------|
| inteiro | meuVetor |            |

# Vetores



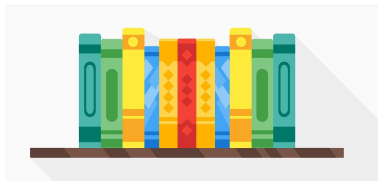
| Tipo   | Nome         | Capacidade |
|--------|--------------|------------|
| livros | minhaEstante | 11         |

livros minhaEstante[11];

|   |    |    |     |   |
|---|----|----|-----|---|
| 1 | 26 | 22 | 100 | 2 |
|---|----|----|-----|---|

| Tipo    | Nome     | Capacidade |
|---------|----------|------------|
| inteiro | meuVetor | 5          |

# Vetores



| Tipo   | Nome         | Capacidade |
|--------|--------------|------------|
| livros | minhaEstante | 11         |

livros minhaEstante[11];

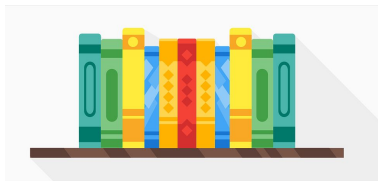
|   |    |    |     |   |
|---|----|----|-----|---|
| 0 | 1  | 2  | 3   | 4 |
| 1 | 26 | 22 | 100 | 2 |

| Tipo | Nome     | Capacidade |
|------|----------|------------|
| int  | meuVetor | 5          |

inteiro meuVetor[5];



# Vetores



| Tipo   | Nome         | Capacidade |
|--------|--------------|------------|
| livros | minhaEstante | 11         |

livros minhaEstante[11];

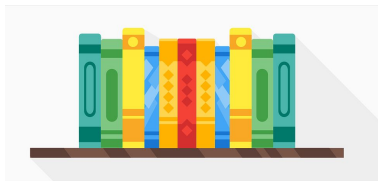
|    |    |    |     |    |
|----|----|----|-----|----|
| 0  | 1  | 2  | 3   | 4  |
| 30 | 26 | 22 | 100 | 40 |

- meuVetor[0] = 30;
- meuVetor[4] = 40;

| Tipo | Nome     | Capacidade |
|------|----------|------------|
| int  | meuVetor | 5          |

inteiro meuVetor[5];

# Vetores



| Tipo   | Nome         | Capacidade |
|--------|--------------|------------|
| livros | minhaEstante | 11         |

livros minhaEstante[11]

|    |    |    |     |    |
|----|----|----|-----|----|
| 0  | 1  | 2  | 3   | 4  |
| 30 | 26 | 50 | 100 | 40 |

- meuVetor[0] = 30
- meuVetor[4] = 40
- meuVetor[2] = 50

| Tipo | Nome     | Capacidade |
|------|----------|------------|
| int  | meuVetor | 5          |

inteiro meuVetor[5]

# Vetores

É uma das estruturas de dados mais simples e mais utilizadas dentre todas.  
Principais características:

- Indexação com início em 0 (zero)
- Adição e pesquisa de novos elementos de forma aleatória
- Acesso aos elementos através de índices
- Possuem tamanho finito de elementos
- Carregam dados de tipos específicos
- Podem possuir uma ou mais dimensões

## Inserindo elementos no vetor

```
programa
{
    funcao inicio()
    {
        cadeia nome[5]
        nome[0] = "Jorge"
        nome[1] = "Ana"
        nome[2] = "Roni"
        nome[3] = "Carla"

        escreva(nome[3])
    }
}
```

O que será exibido se colocarmos nome[4]?

Outras formas de definição

```
programa
{
    funcao inicio()
    {
        cadeia uf[5] = {"RJ", "SP", "MG", "BA", "RS"}

        escreva(uf[0])
    }
}
```

```
programa
{
    funcao inicio()
    {
        cadeia uf[] = {"RJ", "SP", "MG", "BA", "RS", "SE", "CE", "AL", "MA", "PA"}

        escreva(uf[1])
    }
}
```

# Vetores

Em uma variável é guardado somente o último valor conforme o exemplo abaixo na leitura da idade de cinco pessoas.

```
programa
{
    funcao inicio()
    {
        inteiro idade
        para(inteiro i=0; i<4; i++){
            escreva("Sua idade:")
            leia(idade)
        }
    }
}
```

Para armazenar o valor das variáveis teriam que ser criadas cinco variáveis o que pode ser resolvido criando um vetor.

```
programa
{
    funcao inicio()
    {
        inteiro idade[5]

        para(inteiro i=0; i<=4; i++){
            escreva("Sua idade:")
            leia(idade[i])
        }

        para(inteiro i=0; i<=4; i++){
            escreva(idade[i], "\n")
        }
    }
}
```

## Exercício

Altere o algoritmo do slide anterior para descobrir qual a maior e a menor idade

```
programa
{
    funcao inicio()
    {
        inteiro idade[5], maiorIdade=0, menorIdade=0
        para(inteiro i=0; i<=4; i++){
            escreva("Sua idade:")
            leia(idade[i])
        }

        para(inteiro i=0; i<=4; i++){
            escreva(idade[i],"\n")

            se(i==0){
                maiorIdade = idade[i]
                menorIdade = idade[i]
            } senao{
                se (idade[i] > maiorIdade){
                    maiorIdade = idade[i]
                }

                se (idade[i] < menorIdade){
                    menorIdade = idade[i]
                }
            }
        }
        escreva("Maior Idade:",maiorIdade,"\n")
        escreva("Menor Idade:" ,menorIdade)
    }
}
```

## Exercício

Faça um algoritmo com um vetor com 8 numeros inteiros.

Exiba a soma de todos os números e também quantidade de números pares e ímpares?

```
programa
{
    funcao inicio()
    {
        inteiro numero[8] , soma=0, quantPar=0, quantImpar=0
        para(inteiro i=0; i < 8; i++){
            escreva("Informe o número:")
            leia(numero[i])
            soma = soma + numero[i]
            se(numero[i] %2 ==0){
                quantPar++
            }senao{
                quantImpar++
            }
        }
        escreva("A soma dos números é:", soma, "/n")
        escreva("A quantidade de números pares: ", quantPar, "/n")
        escreva("A quantidade de números ímpares: ", quantImpar)
    }
}
```

## Exercício

Faça um algoritmo que leia as cinco vogais e exiba na tela na ordem inversa.

```
programa
{
    funcao inicio()
    {
        caracter vogal[5]
        para(inteiro i=0; i < 5; i++){
            escreva("Digite a vogal:")
            leia(vogal[i])
        }

        para(inteiro i=4; i >= 0; i--){
            escreva(vogal[i], "\n")
        }
    }
}
```



## Exercício

Criar um algoritmo para leitura de quatro notas em um vetor. Calcular a média, exibir a maior nota, a menor nota

```
programa
{
    funcao inicio()
    {
        real maior=0.0, menor=11.0, media, somaNotas=0, notas[4]
        //Entrada de dados
        para(inteiro i=0; i < 4; i++){
            escreva("Digite a nota ", i+1,":")
            leia(notas[i])

            se (notas[i] > maior){
                maior = notas[i]
            }

            se (notas[i] < menor){
                menor = notas[i]
            }
            somaNotas += notas[i]
        }

        //Exibe no console o vetor de notas
        para(inteiro i=0; i < 4; i++){
            escreva(notas[i], "\n")
        }

        media = somaNotas/4
        escreva("Maior nota:", maior, "\n")
        escreva("Menor nota:", menor, "\n")
        escreva("Média:", media)
    }
}
```

Fazer a crítica para que não sejam digitadas notas inferiores a zero ou superiores a 10

Escreva um algoritmo que leia uma lista de salários de cinco funcionários em um vetor. Após, o algoritmo deverá aplicar um aumento de 10% somente sobre salários abaixo de R\$ 2000,00. Mostrar na tela a lista dos salários.

```
programa
{
    funcao inicio()
    {
        real salario[5]

        para(inteiro i = 0; i < 5; i++){
            escreva("\nInforme o ",i+1,"º salário: ")
            leia(salario[i])

            se(salario[i] < 1500){

                salario[i] += salario[i] * 0.1
            }
        }
        escreva("\nLista de salários:")
        para(inteiro i = 0; i < 5; i++){
            escreva("\n",i+1,"º salário é R$ ",salario[i])
        }
    }
}
```

Criar um algoritmo que leia em um vetor o nome, a quantidade e o valor de uma lista de três produtos. Ao final deverá calcular o subtotal de cada produto e no final exibir o total geral da compra

```
programa
{
    funcao inicio()
    {
        real valor[3], subtotal[3], quantidade[3], totalGeral = 0.0
        cadeia nome[3]

        para(inteiro i = 0; i < 3; i++){
            escreva("\nInforme o nome do produto: ")
            leia(nome[i])
            escreva("\nInforme a quantidade do produto: ")
            leia(quantidade[i])
            escreva("\nInforme o valor do produto: ")
            leia(valor[i])
            subtotal[i] = valor[i] * quantidade[i]
            totalGeral += subtotal[i]
            limpa()
        }
        para(inteiro i = 0; i < 3; i++){
            escreva("\nO subtotal do produto ",nome[i]," é de R$ ",subtotal[i])
        }
        escreva("\nO valor total é de R$ ",totalGeral)
    }
}
```

Elabore um algoritmo que leia em um vetor:

- um vetor com os nomes de seis times.
- outro vetor com a pontuação dos seis times.

Exibir ao final o nome do time campeão e o último colocado na pontuação.

```
programa
{
    funcao inicio()
    {
        cadeia time[3], campeao = " ", ultimo = " "
        real pontuacao[3], maior=0.0, menor = 1000.0

        para(inteiro i = 0; i < 3; i++){

            escreva("\nInforme o nome do time: ")
            leia(time[i])
            escreva("\nInforme a pontuação: ")
            leia(pontuacao[i])

            se(pontuacao[i] >= maior){
                maior = pontuacao[i]
                campeao = time[i]
            }

            se(pontuacao[i] <= menor){
                menor = pontuacao[i]
                ultimo = time[i]
            }

        }

        escreva("\nCampeão: ",campeao,". Total de pontuacaos: ",maior,"\n")
        escreva("\nÚltimo: ",ultimo,". Total de pontuacaos: ",menor,"\n")

    }
}
```