

NExT

Nova Experiência de Trabalho

Tatyane Calixto
tscs@cesar.org.br

Erick Simões
esm@cesar.org.br



Os breakout rooms já
estão abertos!

AGEND

A

19:00 - 20:00 Resolução de exercícios

Resolução de questões das listas de exercícios passados

20:00 - 21:00 Aula do dia: Vetor

Estrutura de dados vetor, como criar e acessar seus elementos

21:00 - 22:00 Lista de exercícios

Lista de exercícios com questões que envolvem estruturas de repetição e vetores



REVISÃO

estruturas de repetição:
para

ENQUANTO

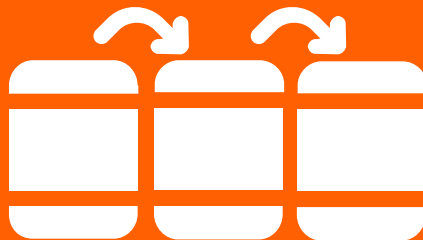
```
enquanto(condição) {  
    //comandos que são executados repetidamente  
}
```

FACA-ENQUANTO

```
faca {  
    //comandos que são executados repetidamente  
} enquanto(condição)
```

PARA

```
para(variável; condição; incremento){  
    //comandos que são executados repetidamente  
}
```



VETOR

Estrutura de Dados

Estruturas de Dados

Em diversos exercícios, nos deparamos com situações em que tínhamos a necessidade de ler **vários dados de um mesmo tipo**.

Na maioria das vezes conseguimos fazer isso acumulando os valores recebidos em variáveis "somatório".

Contudo, isso gera um problema: **perdemos os dados individuais**.

Estruturas de Dados

INTRODUÇÃO

Estruturas de Dados

INTRODUÇÃO

Estruturas de Dados

Quando criamos um programa é comum representarmos informações do mundo real organizadas em estruturas que possam ser manipuladas de forma mais simples.

Essas estruturas, podem conter vários dados de um mesmo tipo em uma única variável, dispostos de forma organizada pela ordem.

O acesso a cada um desses dados pode ser feito através de sua posição.

Vetor

O vetor, também chamado de **array**, é uma **estrutura de dados**. Nele, um conjunto de elementos de um **mesmo tipo** é armazenado em uma única variável.

```
inteiro idade = 27
```

```
inteiro notas[4] = {10, 8, 7, 9}
```



Estruturas de Dados

VETOR

Vetor

Como declarar um vetor:



```
inteiro notas[4]  
caracter vogais[] = {'a', 'e', 'i', 'o', 'u'}  
real alturas[3] = {1.8, 1.5, 2.0}
```

colchetes para o tamanho

chaves para os elementos

O que não é permitido na declaração de um vetor:



```
inteiro notas[4] = {10, 5}  
logico respostas[] = {verdadeiro, falso, 1}
```

Estruturas de Dados

VETOR

Vetor

Como acessar um elemento do vetor

- Acessando a posição diretamente:

```
cadeia liga_da_justica[4] = {"Mulher Maravilha",  
"Flash", "Batman", "Superman"}  
  
escreva(liga_da_justica[1]) // Flash
```

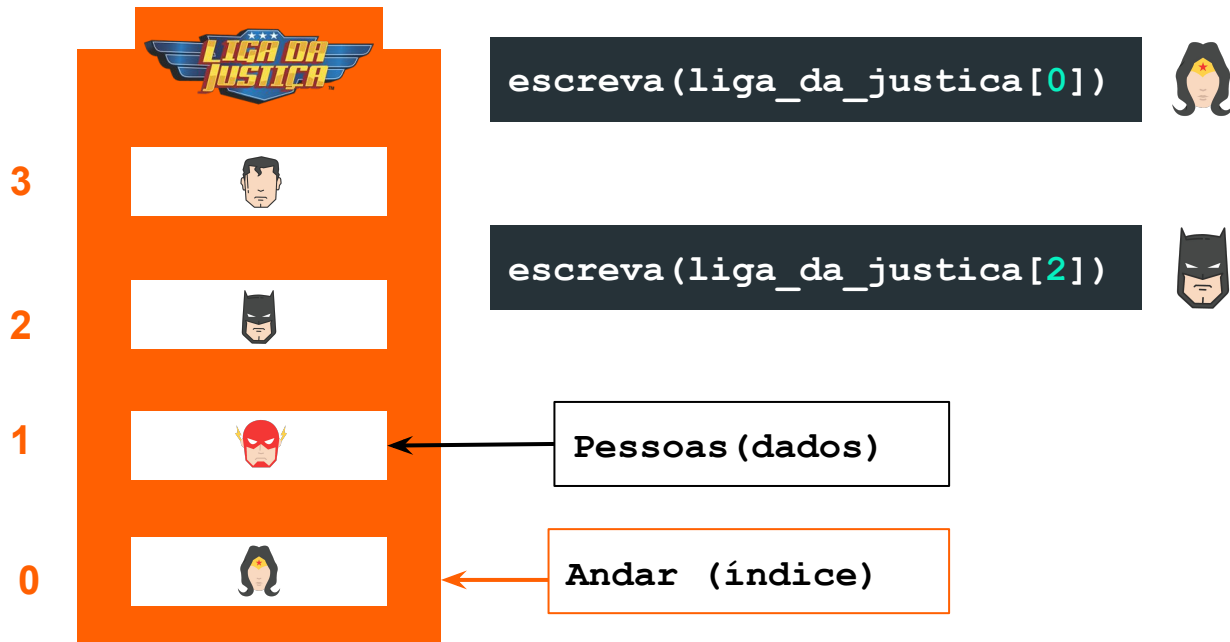
Estruturas de Dados

VETOR

```
cadeia liga_da_justica[4] = {"Mulher Maravilha",  
"Flash", "Batman", "Superman"}
```

Estruturas de Dados

VETOR



Obs: 1 apartamento por andar

Vetor

Como acessar um elemento do vetor

- Usando uma estrutura de repetição:

```
cadeia liga_da_justica[4] = {"Mulher Maravilha",  
"Flash", "Batman", "Superman"}  
  
para (inteiro i=0; i<4; i++) {  
    escreva(liga_da_justica[i], "\n")  
}
```

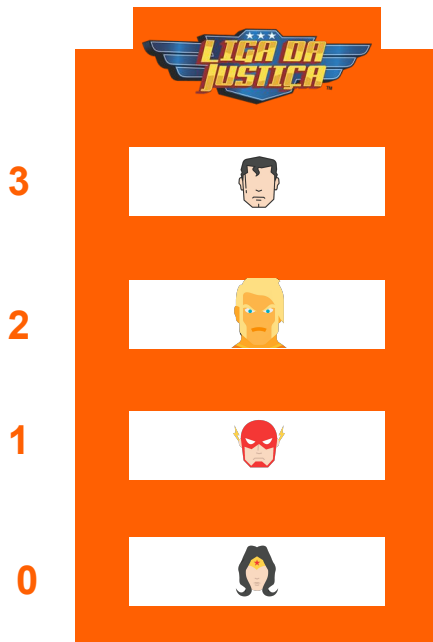
Estruturas de Dados

VETOR

```
cadeia liga_da_justica[4] = {"Mulher Maravilha", "Flash",  
"Batman", "Superman"}  
  
liga_da_justica[2] = "Aquaman" // Batman substituído por Aquaman
```

Estruturas de Dados

VETOR



```
escreva(liga_da_justica[2])
```



Vetor

É possível alterar os valores armazenados em determinada posição do vetor:

```
cadeia liga_da_justica [4] = {"Mulher Maravilha", "Flash",  
"Batman", "Super Homem"}  
  
para(inteiro i=0; i < 4; i++){  
    escreva(liga_da_justica[i], "\n")  
}  
  
liga_da_justica[2] = "Aquaman" //Batman será substituído pelo Aquaman  
  
escreva("\n")  
  
para(inteiro i=0; i < 4; i++){  
    escreva(liga_da_justica[i], "\n")  
}
```

Estruturas de Dados

VETOR



EXEMPLOS

vetor

EXEMPLO 1:

Crie um programa e declare um vetor de valores inteiros com 5 números. Exiba o valor que está:

- na posição inicial;
- na posição mediana;
- na posição final.

Estruturas de Repetição

EXEMPLO: VETOR

EXEMPLO 1:

Crie um programa e declare um vetor de valores inteiros com 5 números. Exiba o valor que está:

- na posição inicial;
- na posição média;
- na posição final.

```
inteiro numeros[] = {2, 15, 20, 4, 7}
```

```
escreva(numeros[0], "\n")
```

```
escreva(numeros[2], "\n")
```

```
escreva(numeros[4], "\n")
```

Estruturas de Repetição

EXEMPLO: VETOR

EXEMPLO 2:

Crie um programa com dois vetores. Um para armazenar os nomes de alguns monitores, outro para armazenar sua altura. Exiba o nome do monitor e sua altura.

Nome	Altura
Isa	1.67
Aleixo	1.71
Borges	1.56
Jorge	1.68
Mar	1.84
Omena	1.70

Estruturas de Repetição

EXEMPLO: VETOR

EXEMPLO 2:

Estruturas de Repetição

EXEMPLO: VETOR

```
cadeia nomes[] = {"Isa", "Aleixo", "Borges",  
"Jorge", "Mar", "Omena"}  
real alturas[] = {1.67, 1.71, 1.56, 1.68, 1.84,  
1.70}  
  
para (inteiro i = 0; i < 6; i++) {  
    escreva(nomes[i], " - ")  
    escreva(alturas[i], "\n")  
}
```

Estruturas de Repetição

EXEMPLO: VETOR

EXEMPLO 3:

Crie um programa usando os vetores anteriores. Nesse novo programa, você deve receber do usuário sua altura e informar o nome do(a) monitor(a) que tem a mesma altura que você. Informe ao usuário se não houver nenhum (a) monitor(a) com a altura igual.

EXEMPLO 3:

```
cadeia nomes[] = {"Isa", "Aleixo", "Borges", "Jorge", "Mar", "Omena"}
real alturas[] = {1.67, 1.71, 1.56, 1.68, 1.84, 1.70}
real altura_usuario
logico encontrado = falso

leia(altura_usuario)

para(inteiro i=0; i<6; i++) {
    se(altura_usuario == alturas[i]){
        escreva("O(a) monitor(a) ", nomes[i], " tem a mesma altura que você")
        encontrado = verdadeiro
    }
}

se(encontrado == falso){
    escreva("Ninguém tem uma altura igual a sua")
}
```

Estruturas de Repetição

EXEMPLO: VETOR

Estruturas de Repetição

EXEMPLO: VETOR

EXEMPLO 4:

Crie um programa que calcule o perímetro de quaisquer figuras planas (soma de todos os lados). O usuário deve informar no início do programa quantos lados a figura possui. É obrigatório que a figura tenha no mínimo 3 lados. Em seguida, o programa deve fazer a leitura do valor de cada um dos lados da figura. No final, exiba o valor da soma de todos os lados.

No Portugol, o tamanho de um vetor só pode ser definido por um valor literal (escrito diretamente no código) ou por uma constante (que precisa ser inicializada).

Desta forma, não é possível criar um vetor com o tamanho informado pelo usuário.

Breakout Time!

Resolva os desafios da lista de exercícios com sua sala no breakout room.

A lista possui exercícios em duas categorias:

- Exercícios fundamentais;
- Exercícios de aprofundamento.

Se precisar de ajuda, chame uma das pessoas monitoras ou professoras.



Lista de Exercícios 07



C . E . S . A . R

Pessoas impulsionando inovação.
Inovação impulsionando negócios.

Tatyane Calixto
tscs@cesar.org.br

Erick Simões
esm@cesar.org.br

e a melhor equipe de monitores
da CESAR School 

