

# NExT

## Nova Experiência de Trabalho

Tatyane Calixto  
[tscs@cesar.org.br](mailto:tscs@cesar.org.br)

Erick Simões  
[esm@cesar.org.br](mailto:esm@cesar.org.br)



C.E.S.A.R



Os breakout rooms já  
estão abertos!

# AGEND

# A

## **19:05 - 19:30** Resolução de exercícios

Resolução de questões das listas de exercícios passados

## **19:30 - 22:00** Lista de exercícios de revisão

Lista de exercícios com questões que envolvem todos os conteúdos já abordados

## **19:05 - 22:00** 21 - Save Room

Sala especial reservada aos estudantes que não estiveram presentes nas aulas ao vivo ou estão sentindo dificuldade com os primeiros conteúdos





# REVISÃO

estruturas de repetição:  
enquanto e faca-enquanto

# ENQUANTO

*Pode não ser executado*

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

# FACA-ENQUANTO

*Certeza que vai ser executado ao menos uma vez*

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



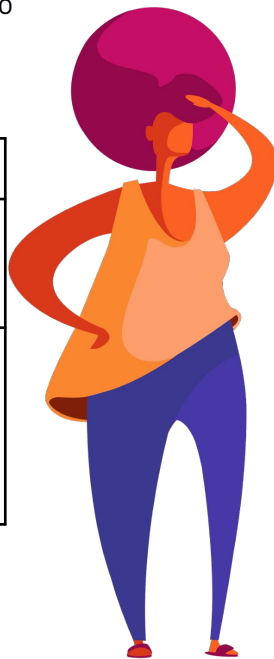
# EXEMPLOS

para

## EXEMPLO 1:

Faça um programa que leia apenas 3 números ímpares e exiba o menor valor, o maior valor e a soma dos valores. Enquanto o usuário continuar digitando números pares o programa continuará pedindo a entrada correta.

Exemplos Entrada	Exemplos Saída
1 5 3	O menor valor é: 1 O maior valor é: 5 A soma dos valores: 9
1 5 2 8 3	O menor valor é: 1 O maior valor é: 5 A soma dos valores: 9



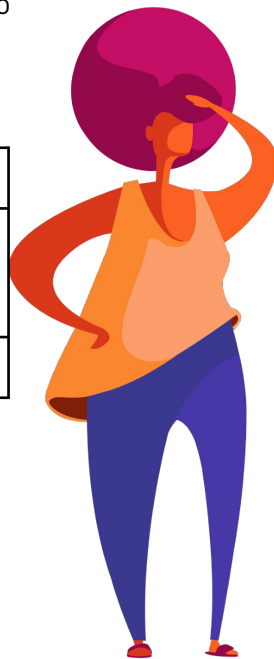
# Estruturas de Repetição

ENQUANTO E  
FACA-ENQUANTO

## EXEMPLO 2:

Faça um programa onde o usuário precisa escolher uma cor de camisa, branca ou preta. Enquanto ele digitar qualquer valor diferente de branco ou preto o programa continuará perguntando.

Exemplos Entrada	Exemplos Saída
azul verde branca	Cor de camisa inválida. Digite uma cor correta. Cor de camisa inválida. Digite uma cor correta. Cor correta.
preto	Cor correta.



# Estruturas de Repetição

ENQUANTO E  
FACA-ENQUANTO



## EXEMPLO 3:

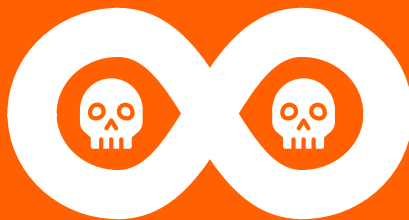
Faça um programa que permita a leitura da média de 5 alunos. Calcule a média da turma e exiba quantos alunos obtiveram nota acima da média da turma calculada.

Exemplos Entrada	Exemplos Saída
5 6 7 8 9	A média da turma é: 7.0. Estudantes que obtiveram a nota maior que a média da turma: 2



## Estruturas de Repetição

ENQUANTO E  
FACA-ENQUANTO



# LOOP INFINITO

Estruturas de Repetição

# Estruturas de Repetição

## LOOP INFINITO

### Loop Infinito

- Ocorre quando a condição de saída do laço nunca é atingida;
- A execução do programa fica presa no bloco do laço;
- Pode causar o consumo excessivo de memória e processamento e, conseqüentemente, o travamento do programa.

# Estruturas de Repetição

## LOOP INFINITO

```
inteiro contador = 0

enquanto(contador < 10) {
    escreva("Contando número: ", contador, "\n")
}
```



```
inteiro contador = 0

faca {
    escreva("Contando número: ", contador, "\n")
} enquanto(contador < 10)
```





# PARA

Estrutura de Repetição

## para

Permite repetir um bloco de código **enquanto** a **expressão no centro for verdadeira**. A primeira expressão declara uma variável que funcionará apenas nesta repetição e a última incrementa a mesma a cada final da repetição.

*executado 1 vez*  
*condição checada a cada repetição*  
*executado a cada repetição*

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

# Estruturas de Repetição

PARA

## para - sintaxe

*de* *para* *passo*

```
para(inteiro i = 0; i < 10; i++) {  
    //comandos que são executados repetidamente  
}
```

# Estruturas de Repetição

PARA

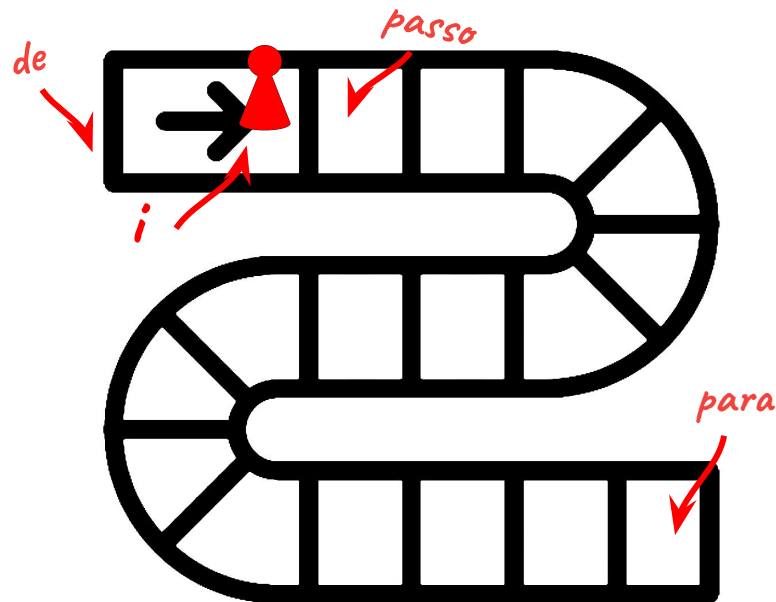
Este laço possui 3 partes:

1. Inicializador com a variável contadora;
2. Definição da condição de parada (valor final do contador);
3. Definição do incremento.

## para - sintaxe

*de* *para* *passo*

```
para(inteiro i=1; i <= 16; i++){  
    //comandos que são executados repetidamente  
}
```



# Estruturas de Repetição

PARA



## para - características

Quando pode ser usado:

- Já se sabe quantas vezes o loop será executado;
- Existe a necessidade de se usar um contador;

```
para (inteiro i = 0; i < 10; i++) {  
    escreva(i, "\n")  
}
```

# Estruturas de Repetição

PARA



# EXEMPLOS

para

## EXEMPLO 4:

Crie um programa que irá perguntar ao usuário quantos números ele deseja inserir. Em seguida, leia todos os  $n$  números e, no final, informe qual o maior e o menor número inserido.



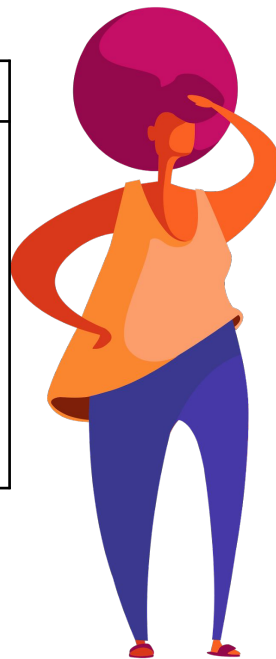
# Estruturas de Repetição

PARA

## EXEMPLO 5:

Faça um programa que dê como saída a tabuada de um número inteiro (entre 1 e 10) recebido pelo console.

Exemplos Entrada	Exemplos Saída
4	$4 \times 1 = 4$ $4 \times 2 = 8$ $4 \times 3 = 12$ $4 \times 4 = 16$ $4 \times 5 = 20$ $4 \times 6 = 24$ $4 \times 7 = 28$ $4 \times 8 = 32$ $4 \times 9 = 36$ $4 \times 10 = 40$



# Estruturas de Repetição

PARA

## EXEMPLO 6:

A prefeitura de uma cidade fez uma pesquisa entre seus 10 funcionários, coletando dados sobre o salário e número de filhos. A prefeitura deseja saber:

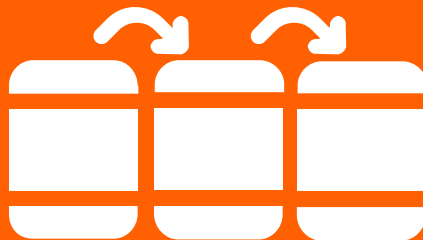
- a) Média do salário da população;
- b) Média do número de filhos;
- c) Maior salário;
- d) Percentual de pessoas com salário até R\$ 1000,00.

Exemplos Entrada (apenas 3 funcionários)	Exemplos Saída
1200 2 900 1 850 2	Média salarial: 983.33333 Média do número de filhos: 1.66666 Maior salário:1200 66.66666% pessoas têm salários até R\$ 1000,00



# Estruturas de Repetição

PARA



# 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

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.

# Estruturas de Dados

## INTRODUÇÃO



## 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:

*colchetes para o tamanho*

*chaves para os elementos*



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

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
```

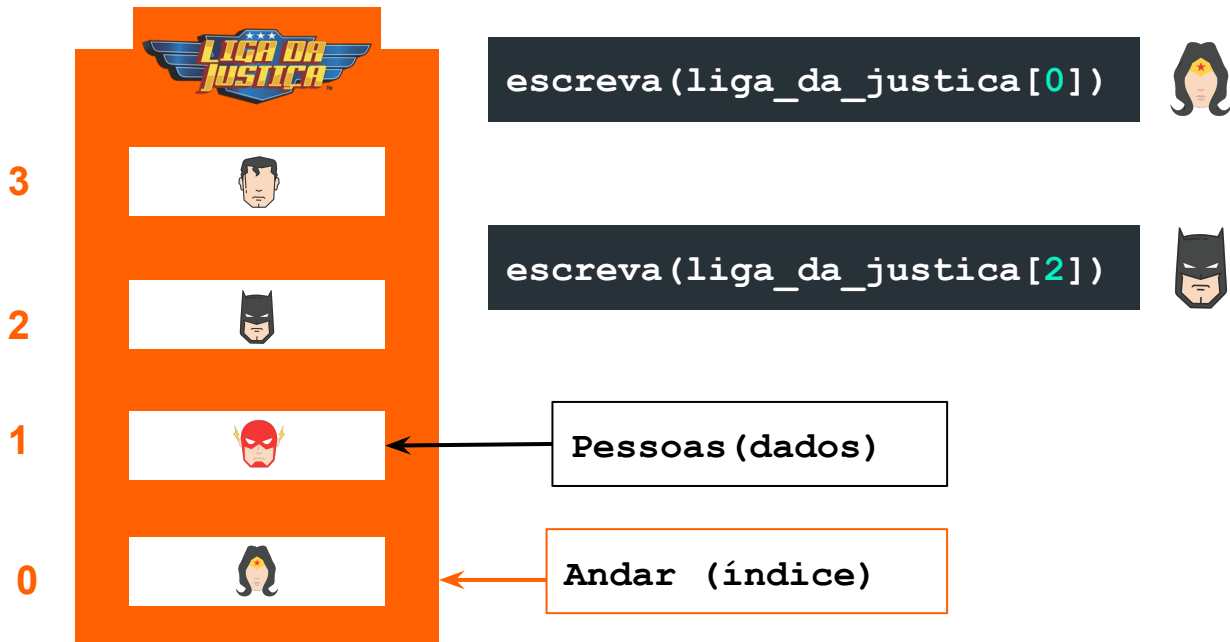
# Estruturas de Dados

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



# EXEMPLOS

para

## EXEMPLO 7:

Receba do usuário 5 números. Em seguida, exiba-os em ordem inversa (em relação à ordem em que foram inseridos)

Exemplos Entrada	Exemplos Saída
1	2
4	0
-7	-7
0	4
2	1



# Estruturas de Repetição

VETOR

## EXEMPLO 8:

Receba as notas de 5 estudantes e as armazene em um vetor. Em seguida, exiba quantos desses estudantes foram aprovados (nota maior ou igual a 7), quantos vão ser reprovados (nota menor que 7) e quantos estão em estado crítico (nota inferior a 3).



# Estruturas de Repetição

VETOR



## EXEMPLO 9:

Crie um programa com dois vetores, cada um com 5 elementos, um deles irá armazenar os nomes dos estudantes, outro irá armazenar suas respectivas idades. Durante a execução do programa, peça que o usuário informe os nomes e idades de cada um dos estudantes. No final, exiba a relação estudante/idade. Exemplo de saída (com 3 elementos apenas):

**Predo - 30 anos**

**Lucas - 25 anos**

**Triago - 29 anos**



# Estruturas de Repetição

VETOR

## EXEMPLO 10:

Estudantes de uma determinada escola fizeram uma prova com 5 questões de verdadeiro ou falso. Crie um programa que recebe as respostas do estudante e no final diz a nota dele, levando em consideração o seguinte gabarito:

Nº da questão	Resposta
1	verdadeiro
2	falso
3	falso
4	verdadeiro
5	verdadeiro

Exemplos Entrada	Exemplos Saída
verdadeiro verdadeiro falso verdadeiro verdadeiro	Você acertou 4 de 5 questões



# Estruturas de Repetição

VETOR

# Acabou \o/





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 

