

# NExT

## Nova Experiência de Trabalho

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

Prof. Erick Simões – [erick.simoaes@cesar.org.br](mailto:erick.simoaes@cesar.org.br)



C . E . S . A . R





# REVISÃO

ou quase isso...

# PRÁTICA 1:

Faça um algoritmo para ler 3 números inteiros e informar a média entre eles.

```
inteiro n1, n2, n3  
leia(n1, n2, n3)  
  
escreva ("Média: ", (n1 + n2 + n3)/3.0)
```

# PRÁTICA 2:

Faça um algoritmo para ler 100 números inteiros e informar a média entre eles.

```
inteiro n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16, n17, n18, n19, n20, n21,
n22, n23, n24, n25, n26, n27, n28, n29, n30, n31, n32, n33, n34, n35, n36, n37, n38, n39, n40, n41,
n42, n43, n44, n45, n46, n47, n48, n49, n50, n51, n52, n53, n54, n55, n56, n57, n58, n59, n60, n61,
n62, n63, n64, n65, n66, n67, n68, n69, n70, n71, n72, n73, n74, n75, n76, n77, n78, n79, n80, n81,
n82, n83, n84, n85, n86, n87, n88, n89, n90, n91, n92, n93, n94, n95, n96, n97, n98, n99, n100

leia(n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16, n17, n18, n19, n20, n21,
n22, n23, n24, n25, n26, n27, n28, n29, n30, n31, n32, n33, n34, n35, n36, n37, n38, n39, n40, n41,
n42, n43, n44, n45, n46, n47, n48, n49, n50, n51, n52, n53, n54, n55, n56, n57, n58, n59, n60, n61,
n62, n63, n64, n65, n66, n67, n68, n69, n70, n71, n72, n73, n74, n75, n76, n77, n78, n79, n80, n81,
n82, n83, n84, n85, n86, n87, n88, n89, n90, n91, n92, n93, n94, n95, n96, n97, n98, n99, n100)

escreva ("Média: ", (n1 + n2 + n3 + n4 + n5 + n6 + n7 + n8 + n9 + n10 + n11 + n12 + n13 + n14 + n15 +
n16 + n17 + n18 + n19 + n20 + n21 + n22 + n23 + n24 + n25 + n26 + n27 + n28 + n29 + n30 + n31 + n32 +
n33 + n34 + n35 + n36 + n37 + n38 + n39 + n40 + n41 + n42 + n43 + n44 + n45 + n46 + n47 + n48 + n49 +
n50 + n51 + n52 + n53 + n54 + n55 + n56 + n57 + n58 + n59 + n60 + n61 + n62 + n63 + n64 + n65 + n66 +
n67 + n68 + n69 + n70 + n71 + n72 + n73 + n74 + n75 + n76 + n77 + n78 + n79 + n80 + n81 + n82 + n83 +
n84 + n85 + n86 + n87 + n88 + n89 + n90 + n91 + n92 + n93 + n94 + n95 + n96 + n97 + n98 + n99 +
n100)/100.0)
```



# Estruturas de Repetição

Haviam dois cachorros, o pet e  
o repet...

## Imagine as seguintes situações:

- Você precisa fazer um mesmo procedimento uma quantidade específica de vezes;
- Você precisa repetir um procedimento centenas de vezes;
- Você precisa repetir um determinado procedimento até que uma condição seja atendida.



# Estruturas de Repetição

## INTRODUÇÃO

As estruturas de repetição são usadas para controlar a execução de códigos, fazendo ele repetir um determinado bloco enquanto uma condição é verdadeira. Elas também podem ser chamadas de LOOPS ou LAÇOS.

Com estruturas de repetição você pode:

- Repetir trechos de código;
- Reduzir linhas de código;
- Tornar o programa mais robusto e flexível;

No Portugal, existem 3 tipos de laços de repetição: **enquanto**, **faca-enquanto** e **para**.

# Estruturas de Repetição

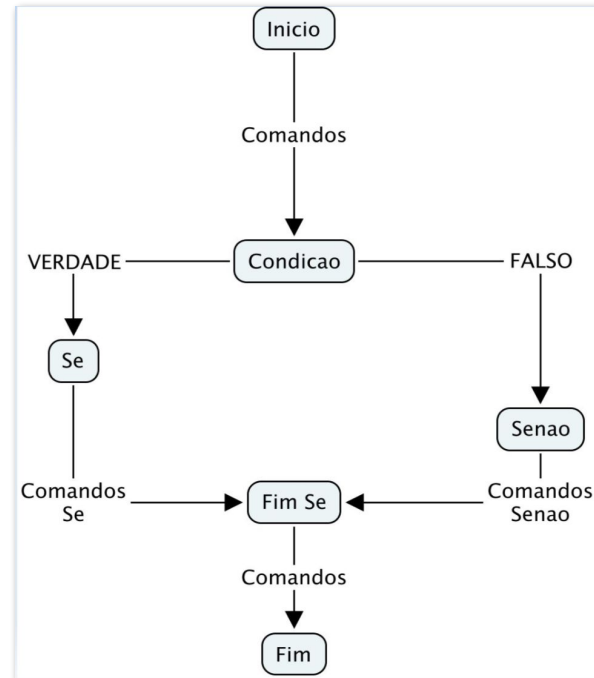
## INTRODUÇÃO

## Fluxo de execução

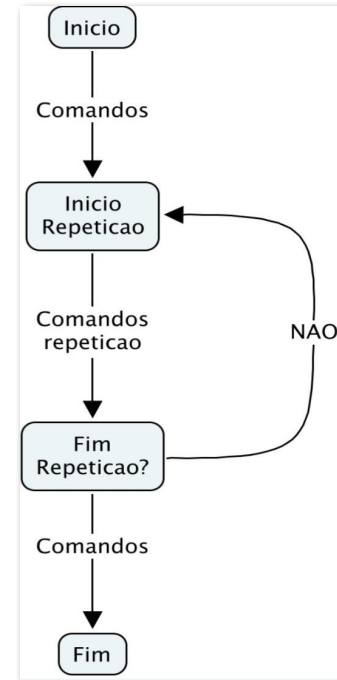
### Normal



### Condicional



### Repetição



# Estruturas de Repetição

## INTRODUÇÃO





# ENQUANTO

Estrutura de Repetição

## enquanto

Enquanto uma condição for verdadeira, repete determinado bloco.

# Estruturas de Repetição

ENQUANTO

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

# Estruturas de Repetição

## ENQUANTO

### enquanto

- Valida a condição logo no início, podendo não executar seu bloco de código;
- Se a condição for verdadeira, executa todos os comandos do bloco e, no final, reavalia a condição;
- Se a condição for falsa, sai do laço e continua o programa.

```
inteiro contador = 0

enquanto (contador < 10) {
    escreva ("Contando número: ", contador)
    contador = contador + 1
}
```

## enquanto

Exemplo 1: Faça um algoritmo para ler 100 números inteiros e informar a média entre eles.

```
inteiro n, somatorio = 0, contador = 0

enquanto(contador < 100) {
    escreva("Insira um número: ")
    leia(n)
    somatorio = somatorio + n
    contador = contador + 1
}

escreva(somatorio / 100.0)
```

# Estruturas de Repetição

ENQUANTO

## enquanto

Exemplo 2: Peça para o usuário informar uma senha com, no mínimo, 8 caracteres. Enquanto essa restrição não for atendida, o programa deve continuar solicitando ao usuário que insira um novo valor.

```
cadeia senha
escreva ("Informe uma senha segura: ")
leia(senha)

enquanto(txt.numero_caracteres(senha) < 8) {
    escreva ("Senha inválida\nTente novamente: ")
    leia(senha)
}
```

# Estruturas de Repetição

ENQUANTO



# PRÁTICA

enquanto

# PRÁTICA 1:

Faça 3 programas:

1. Mostre todos os números de 1 a 100 na tela;
2. Mostre todos os números de 100 a 1 na tela;
3. Mostre todos os números **pares** de 1 a 100 na tela.



[ 🕒 10 minutos ]

## Estruturas de Repetição

### PRÁTICA: ENQUANTO

# PRÁTICA 1.1:

Mostre todos os números de 1 a 100 na tela

```
inteiro contador = 1

enquanto (contador <= 100) {
    escreva (contador, "\n")
    contador = contador + 1
}
```

## Estruturas de Repetição

PRÁTICA: ENQUANTO



# PRÁTICA 1.2:

Mostre todos os números de 100 a 1 na tela;

```
inteiro contador = 100

enquanto (contador > 0) {
    escreva (contador, "\n")
    contador = contador - 1
}
```

## Estruturas de Repetição

PRÁTICA: ENQUANTO

# PRÁTICA 1.3:

Mostre todos os números pares de 1 a 100 na tela.

```
inteiro contador = 1

enquanto (contador <= 100) {
    se (contador % 2 == 0) {
        escreva (contador, "\n")
    }
    contador = contador + 1
}
```

```
inteiro contador = 2

enquanto (contador <= 100) {
    escreva (contador, "\n")
    contador = contador + 2
}
```

## Estruturas de Repetição

PRÁTICA: ENQUANTO

## PRÁTICA 2:

Crie um programa que leia continuamente números inteiros informados pelo usuário, até que um número negativo seja informado. Ao final da leitura mostre a soma dos números positivos digitados.

# Estruturas de Repetição

PRÁTICA: ENQUANTO

[  5 minutos ]

## PRÁTICA 2:

Crie um programa que leia continuamente números inteiros informados pelo usuário, até que um número negativo seja informado. Ao final da leitura mostre a soma dos números positivos digitados.

```
inteiro n, somatorio = 0
leia(n)

enquanto (n >= 0) {
    somatorio += n
    leia(n)
}

escreva(somatorio)
```

# Estruturas de Repetição

PRÁTICA: ENQUANTO

## PRÁTICA 3:

Escreva um algoritmo onde o usuário é solicitado a inserir um nome. Em seguida, ele deve retornar a mensagem:

**Bom dia [nome]!**

O programa deve continuar solicitando que o usuário insira o nome. A condição de saída é o usuário informar o texto **"sair"**.

# Estruturas de Repetição

## PRÁTICA: ENQUANTO

[  5 minutos ]

## PRÁTICA 3:

Escreva um algoritmo onde o usuário é solicitado a inserir um nome. Em seguida, ele deve retornar a mensagem: **"Bom dia [nome]"**. O programa deve continuar solicitando que o usuário insira o nome. A condição de saída é o usuário informar o texto **"sair"**.

```
cadeia nome
escreva("Insira seu nome: ")
leia(nome)

enquanto(nome != "sair") {
    escreva("Bom dia ", nome, "\n")
    escreva("Insira seu nome: ")
    leia(nome)
}
```

# Estruturas de Repetição

PRÁTICA: ENQUANTO



# FACA-ENQUANTO

Estrutura de Repetição

## faca-enquanto

Executa determinado bloco e, **enquanto** uma condição for verdadeira, repete.

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

# Estruturas de Repetição

FACA-ENQUANTO



## faca-enquanto

- **Valida a condição no final, garantindo que o bloco será executado ao menos uma vez;**
- Se a condição for verdadeira, executa todos os comandos do bloco e, no final, reavalia a condição;
- Se a condição for falsa, sai do laço e continua o programa.

```
inteiro contador = 0

faca {
    escreva("Contando número: ", contador, "\n")
    contador = contador + 1
} enquanto(contador < 10)
```

# Estruturas de Repetição

FACA-ENQUANTO

## faca-enquanto

Exemplo 3: Peça para o usuário informar uma senha com, no mínimo, 8 caracteres. Enquanto essa restrição não for atendida, o programa deve continuar solicitando ao usuário que insira um novo valor.

```
cadeia senha

faca {
    escreva ("Informe uma senha segura: ")
    leia(senha)
} enquanto(txt.numero_caracteres(senha) < 8)
```

# Estruturas de Repetição

FACA-ENQUANTO



# PRÁTICA

faca-enquanto

# PRÁTICA 4:

Refaça as práticas anteriores usando o faca-enquanto.

## Estruturas de Repetição

PRÁTICA:  
FACA-ENQUANTO



# 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 05



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 

