

CMPALGP – Algoritmos e Programação



Professores

- Zady Castaneda
zcsalazar318@ifsp.edu.br

Agenda

Estrutura de repetição

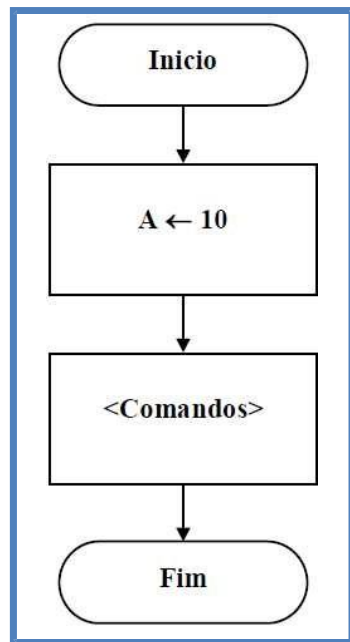
Para

Enquanto

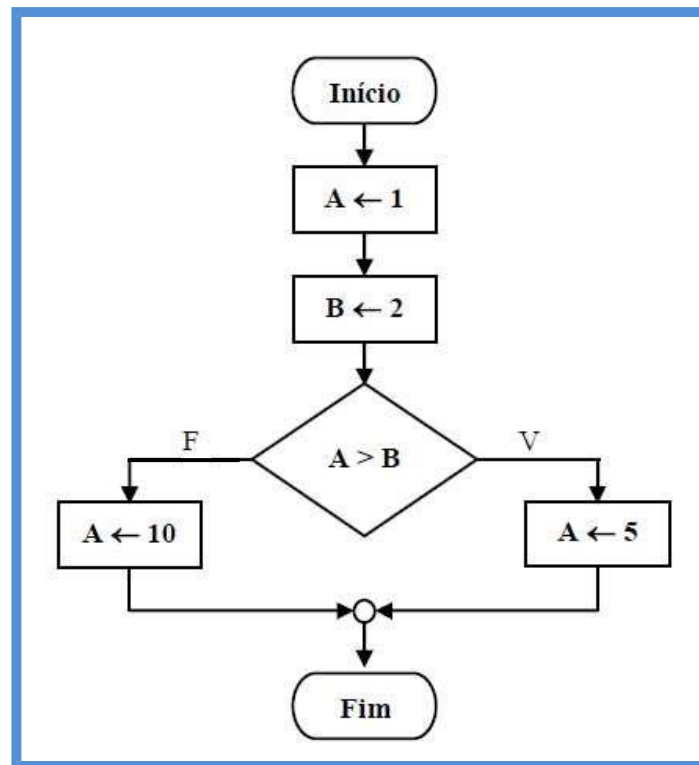
Faca...enquanto

Estruturas de repetição

Existem três estruturas básicas para a construção de algoritmos,
que são:



☐ Linear/ Sequencial



☐ Decisão



☐ Repetição

Estruturas de repetição

- Comando que repete uma sequencia de instruções (comandos) **N** vezes, **até que sua condição (teste) não seja mais verdadeira.**
- As estruturas de repetição também são chamadas de **laços** ou **loops**.

Estruturas de repetição

- São formadas por três partes:
 - **INICIALIZAÇÃO;**
 - **CONDIÇÃO;**
 - **ATUALIZAÇÃO (incremento/decremento);**

- As três partes estão presente em todas as estruturas de repetição, só que em locais diferentes.

Estrutura: **PARA**

- Possui como condição uma **variável de controle**, ou seja, com esta estrutura é possível repetir um conjunto de instruções **até um determinado número de vezes**
- A **variável de controle** é chamada de “**contador**”.
- Recomenda-se utilizar o **PARA** quando as quantidades de repetições forem definidas anteriormente.

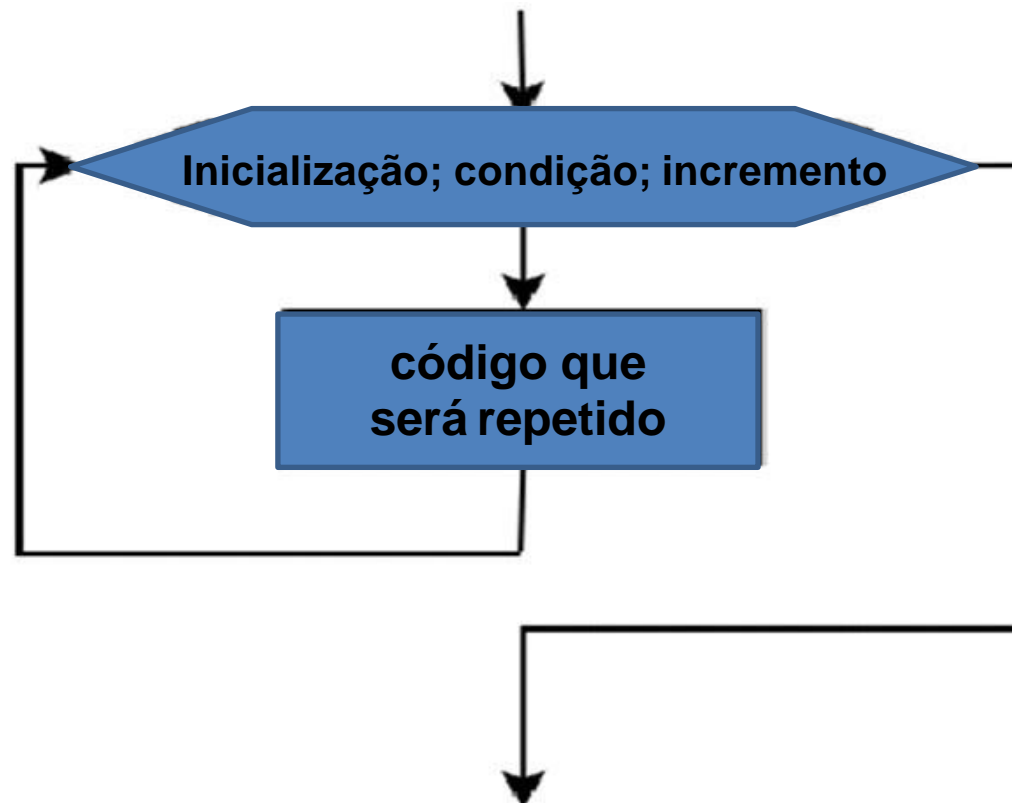
Estrutura: PARA

Sintaxe:

```
para (INICIALIZAÇÃO ; CONDIÇÃO ; ATUALIZAÇÃO)
{
    // código que será repetido
}
```

Estrutura: **PARA**

Fluxograma:



Estrutura: **PARA**

Exemplo:

```
para(inteiro i = 1; i <= 10 ; i++){  
    // Código que será repetido 10 vezes  
}  
...
```

Utilizar **PARA** quando as quantidades de repetições forem definidas anteriormente.

Estrutura: PARA

Exemplo:

```
para(inteiro i = 1; i <= 10; i++){
```

// Código que será repetido 10 vezes

```
}
```

...

INICIALIZAÇÃO

CONDIÇÃO

**INCREMENTO
ou
DECREMENTO**

Estrutura: PARA

- Faça um programa que solicite o número da tabuada (multiplicador) e até quando a tabuada deve ser executada (multiplicando). Apresente as multiplicações (produto) conforme dados informados pelo teclado.

□ Ex: número da tabuada :2

até quando a tabuada vai ser executada: 15

SAIDA

$$2 * 1 = 1$$

$$2 * 2 = 4$$

....

$$2 * 15 = 30$$

programa

```
programa
{
    funcao inicio()
    {
        inteiro numerotabuada, ultimnumero, resultado, contador

        escreva("diga o número da tabuada: ")
        leia(numerotabuada)

        escreva("até quando a tabuada deve ser executada: ")
        leia(ultimnumero)

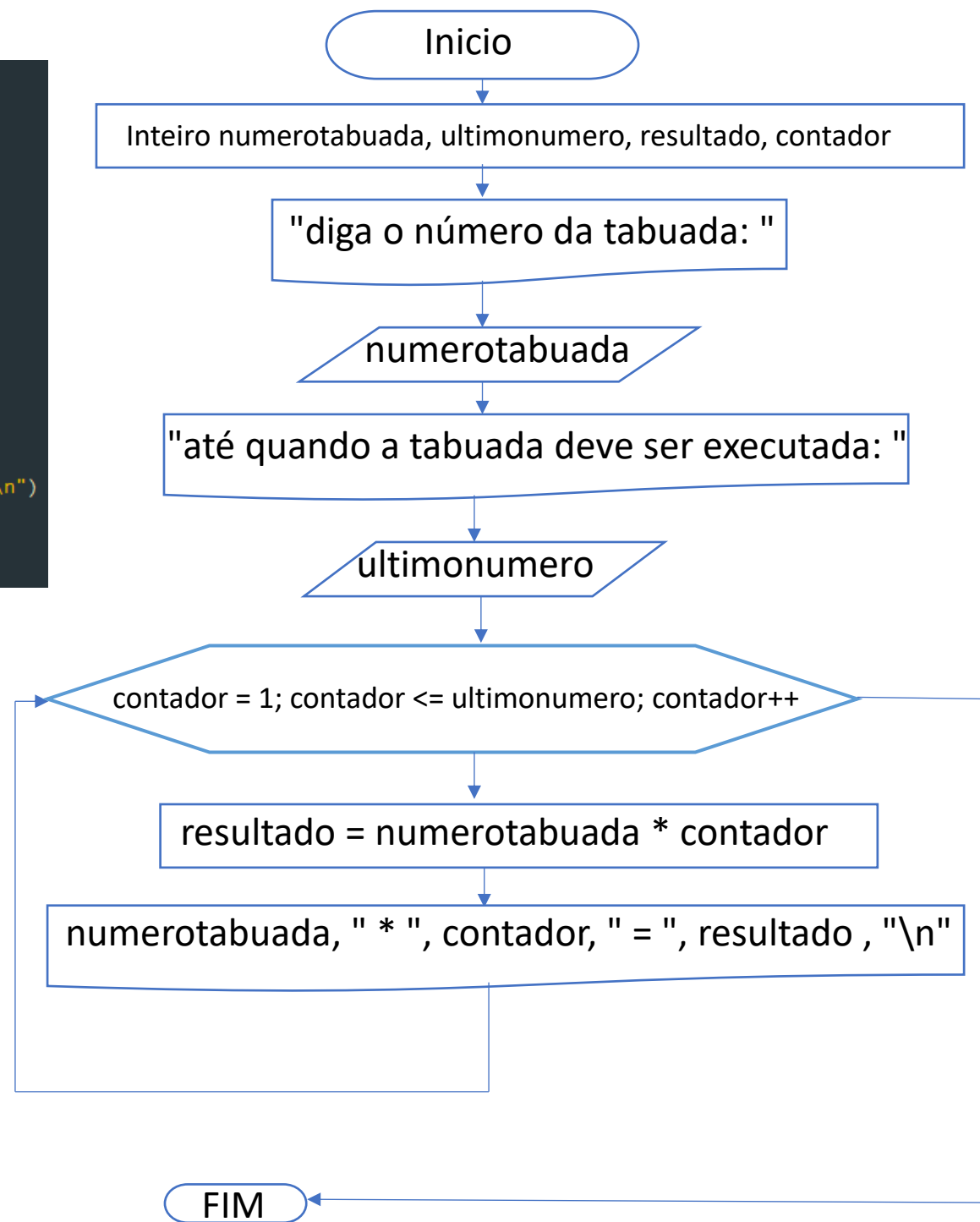
        limpa() // limpa a tela

        para (contador = 1; contador <= ultimnumero; contador++)
        {
            resultado = numerotabuada * contador
            escreva (numerotabuada, " * ", contador, " = ", resultado , "\n")
        }
    }
}
```

SAÍDA

```
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
```

fluxograma



Estrutura: **PARA**

1. Faça um programa que solicite o multiplicador e apresente a multiplicação do número 50 até 0.

Ex: digite o multiplicador: 3

Saída :

3 x 50 =150

3 X 49 =147

....

3 X 0 = 0

2. Faça um programa que apresente os múltiplos de 5 até o número 100.

Ex: 5, 10, 15, 20,..... 100

Estrutura: PARA

3. Faça um algoritmo que receba um número inteiro entre 1 e 7 e mostre na tela o nome do dia da semana associado (por exemplo: 1 = Segunda-feira).
- Caso o número não esteja no intervalo pedido, o programa deve mostrar uma mensagem de erro.
 - O programa deve rodar 10 vezes.
 - A saída deve ser apresentada conforme abaixo: Ex: 1 = Segunda-feira
4 = Quinta-feira

Estrutura: PARA

4. Escreva um algoritmo que receba a idade de 15 pessoas. O algoritmo deve imprimir:
- O total de pessoas com menos de 20 anos.
 - O total de pessoas com mais de 50 anos.
 - A soma de todas as idades.

A saída deve ser apresentada conforme exemplo abaixo:

Total de pessoas com menos de 20 anos = 2

Total de pessoas com mais de 50 anos = 10

Soma de todas as idades = 632

Estrutura: ENQUANTO

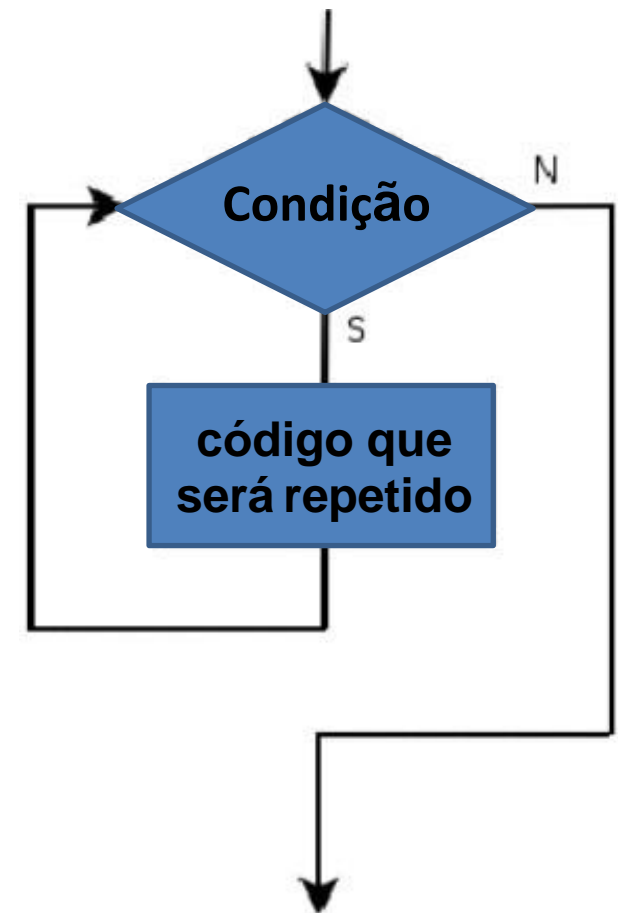
- A repetição é finalizada quando o teste é Falso (F), ou seja, enquanto o teste for Verdadeiro as instruções serão executadas.
- As instruções podem não ser executadas nenhuma vez, pois o teste(condição) fica no início da repetição.

Estrutura: ENQUANTO

Sintaxe:

```
enquanto (condição) {  
    // código que será repetido  
}
```

Fluxograma



Estrutura: ENQUANTO

Exemplo:

```
enquanto (numero < 10) {  
    // código que será repetido  
}
```

Estrutura: ENQUANTO

?

← INICIALIZAÇÃO

enquanto (numero < 10) { ← CONDIÇÃO

// código que será repetido

?

← INCREMENTO
DECREMENTO

}

Estrutura: ENQUANTO

Exemplo:

inteiro numero = 0 ← INICIALIZAÇÃO

enquanto (numero < 10) { ← CONDIÇÃO

// código que será repetido

numero++

← INCREMENTO

}

Estrutura: ENQUANTO

- Faça um programa que solicite o número da tabuada (multiplicador) e até quando a tabuada deve ser executada (multiplicando). Apresente as multiplicações (produto) conforme dados informados pelo teclado.

Ex: número da tabuada : 2

até quando a tabuada vai ser executada: 15

SAIDA

$$2 * 1 = 1$$

$$2 * 2 = 4$$

....

$$2 * 15 = 30$$

programa

```
programa
{
    funcao inicio()
    {
        inteiro numerotabuada, ultimnumero, resultado, contador

        escreva("diga o número da tabuada: ")
        leia(numerotabuada)

        escreva("até quando a tabuada deve ser executada: ")
        leia(ultimnumero)

        limpa() // limpa a tela

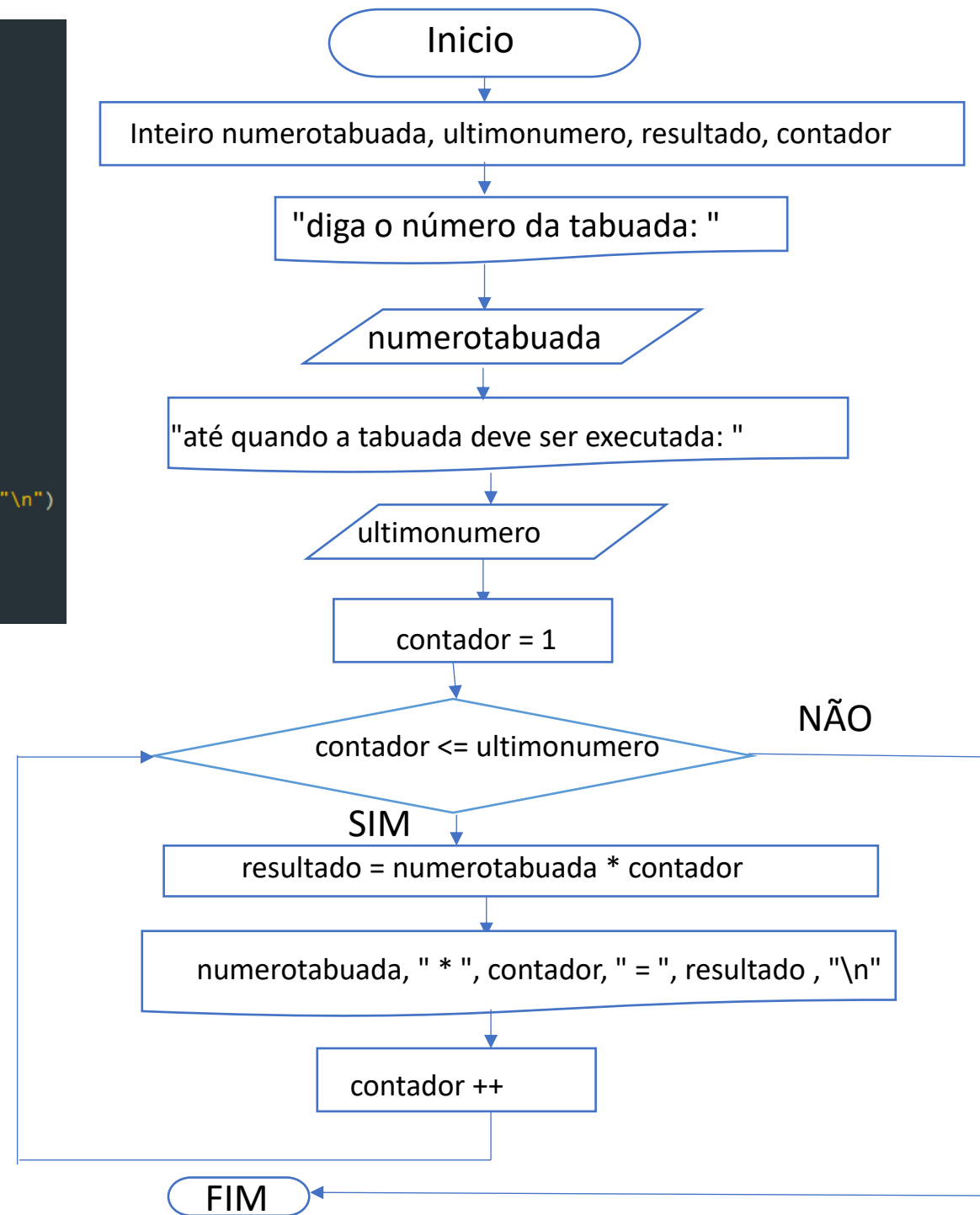
        contador = 1

        enquanto (contador <= ultimnumero)
        {
            resultado = numerotabuada * contador
            escreva (numerotabuada, " * ", contador, " = ", resultado , "\n")
            contador++
        }
    }
}
```

SAÍDA

```
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
```

fluxograma



Estrutura: ENQUANTO

- 5) Entrar com N números até que seja digitado um **número ≤ 0** . Apresente a quantidade de números digitados e a médias deles.
- 6) Faça um programa que leia N número, até que seja digitado um **número ≤ 0** . Apresente para cada número se ele é primo ou não.

Estrutura: **FACA...ENQUANTO**

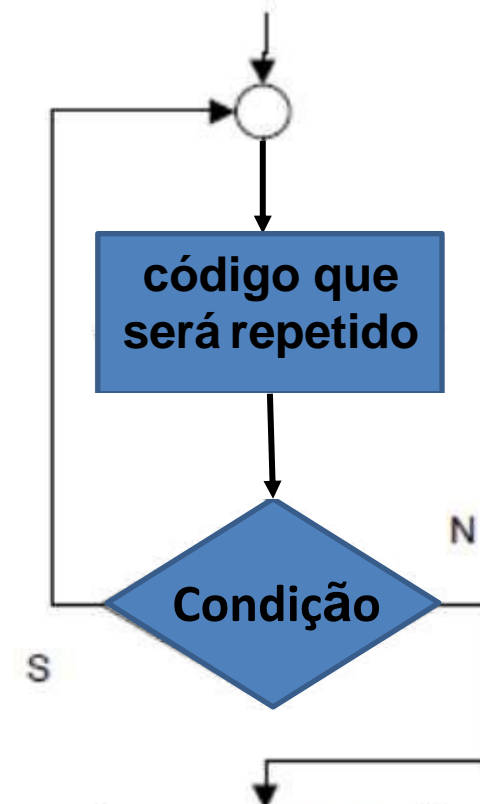
- A repetição é finalizada quando o teste é **Falso (F)**, ou seja, **enquanto o teste for Verdadeiro as instruções serão executadas.**
- As instruções sempre serão executadas pelo menos uma vez, pois o teste(condição) fica no final da repetição.

Estrutura: **FACA...ENQUANTO**

Sintaxe:

```
faca {  
    // código que será repetido  
} enquanto (condição)
```

Fluxograma



Estrutura: **FACA...ENQUANTO**

?

← INICIALIZAÇÃO

faca {

// código que será repetido

?

← INCREMENTO / DECREMENTO

} enquanto (condição)

← CONDIÇÃO

Estrutura: **FACA...ENQUANTO**

Exemplo:

```
inteiro numero = 0
```

```
faca {
```

```
    numero++
```

```
} enquanto (numero < 10)
```

Estrutura: **FACA...ENQUANTO**

Exemplo:

inteiro numero = 0

INICIALIZAÇÃO

faca {

numero++

**INCREMENTO /
DECREMENTO**

} enquanto (**numero < 10**)

CONDIÇÃO

Estrutura: **FACA...ENQUANTO**

- Faça um programa que solicite o número da tabuada (multiplicador) e até quando a tabuada deve ser executada (multiplicando). Apresente as multiplicações (produto) conforme dados informados pelo teclado.

□ Ex: número da tabuada : 2

até quando a tabuada vai ser executada: 15

SAIDA

$$2 * 1 = 1$$

$$2 * 2 = 4$$

....

$$2 * 15 = 30$$

programa

```
programa
{
    funcao inicio()
    {
        inteiro numerotabuada, ultimnumero, resultado, contador

        escreva("diga o número da tabuada: ")
        leia(numerotabuada)

        escreva("até quando a tabuada deve ser executada: ")
        leia(ultimnumero)

        limpa() // limpa a tela

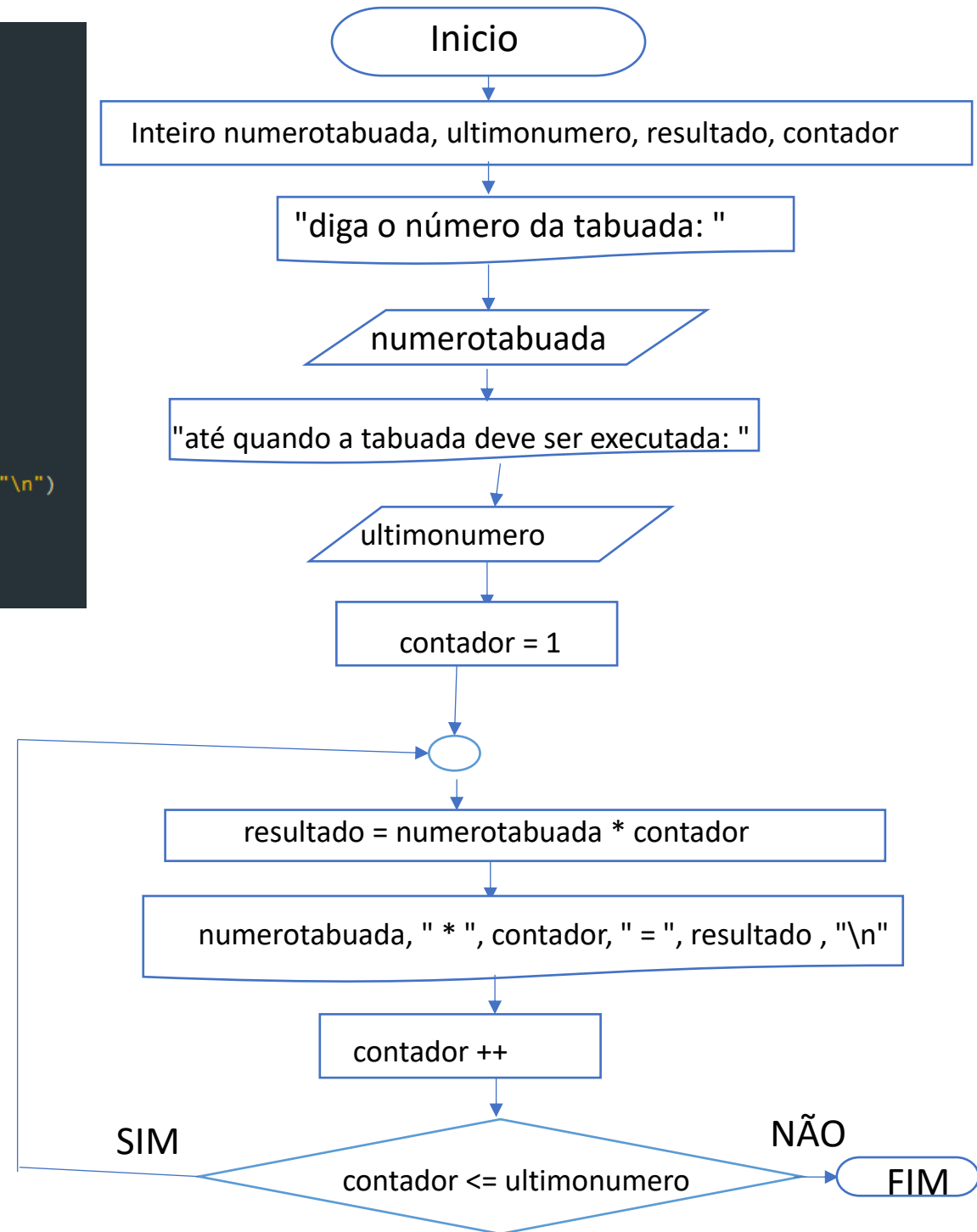
        contador = 1

        faca{
            resultado = numerotabuada * contador
            escreva (numerotabuada, " * ", contador, " = ", resultado , "\n")
            contador++
        }enquanto (contador <= ultimnumero)
    }
}
```

SAÍDA

```
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
```

fluxograma



Estrutura: **FACA...ENQUANTO**

☐ Utilização: Exibir menu

```
programa{  
    funcao inicio(){  
        inteiro menu  
        faca{  
            escreva(" MENU\n")  
            escreva ("1 – Somar\n")  
            escreva ("2 – subtrair \n")  
            escreva("0 – Sair\n")  
            escreva ("Digite sua opção: ")  
            leia (menu)  
            .....  
        }enquanto(menu != 0)  
    }  
}
```

Estrutura: **FACA...ENQUANTO**

□ Utilização: Validação de dados de entrada

```
programa
{
    funcao inicio(){
        inteiro numero
        faca{
            escreva ("Digite um número negativo")
            leia (numero)
        }enquanto (numero > 0)
        escreva ("Parabéns !!! Você digitou um número positivo")
    }
}
```


Estrutura: **FACA...ENQUANTO**

7) Escreva um algoritmo que receba a idade de várias pessoas e pare a execução com a entrada de um número não positivo. O algoritmo deve imprimir:

- O total de pessoas com menos de 21 anos;
- O total de pessoas com mais de 50 anos.

8) Crie um programa para ler a altura de N atletas diferentes. Apresente a média das alturas e a quantidade de atletas com altura superior a **1.5 metros**.

9) Crie um algoritmo para ler o preço de 10 produtos diferentes. Apresente o valor do produto mais barato e mais caro.
Valide a entrada dos valores digitados.

Referências

- ❑ Livro Algoritmos Programação para iniciantes – Gilvan Vilarim. Editora Ciência Moderna
- ❑ Apostila professor Diego Fernandes
- ❑ Material de aula professora Zady e professor Rafael Muniz
- ❑ Google imagens