

# Programação Básica de Computadores

Jordana S. Salamon  
[jssalamon@inf.ufes.br](mailto:jssalamon@inf.ufes.br)

DEPARTAMENTO DE INFORMÁTICA  
CENTRO TECNOLÓGICO  
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

# Estrutura de Repetição - Loops



nemo

# Definição

Loop é uma palavra inglesa, que originalmente significa “aro”, “anel” ou “sequência”.

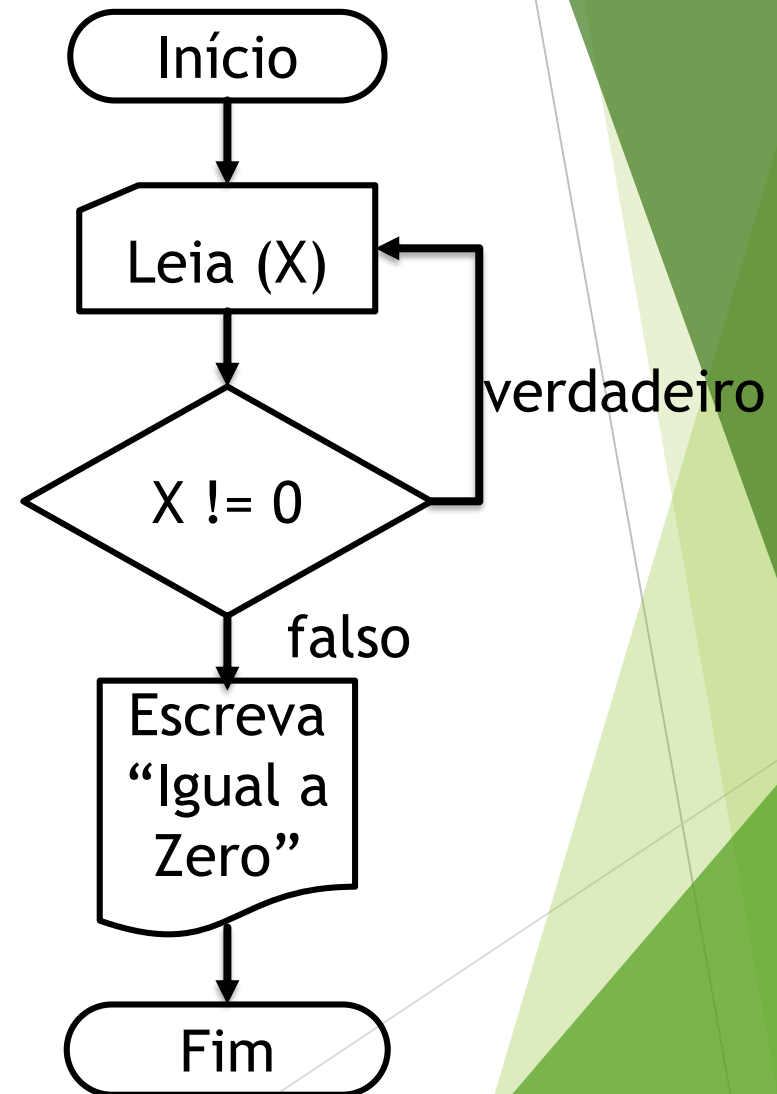
**Exemplo:** Faça o fluxograma que resolva os seguintes conjuntos de passos:

1. Leia um número
2. Verifique se o número é diferente de zero.
3. Caso seja retorne ao passo 1
4. Caso não seja imprima “Igual a Zero”

# Definição

**Exemplo:** Faça o fluxograma que resolva os seguintes conjuntos de passos:

1. Leia um número
2. Verifique se o número é diferente de zero.
3. Caso seja retorne ao passo 1
4. Caso não seja imprima “Igual a Zero”



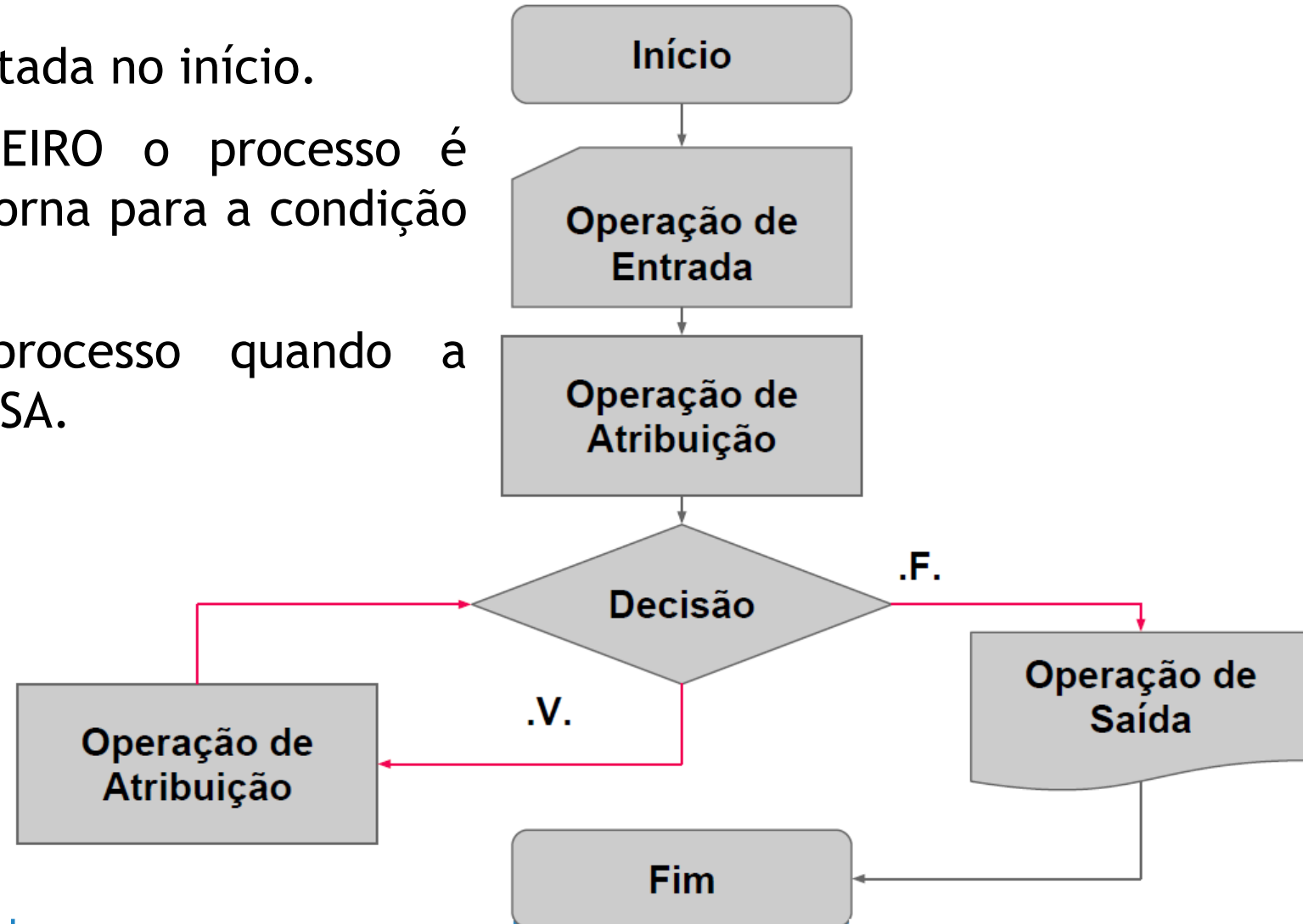
# Tipos de Laços de Repetição

- ▶ Existem dois tipos de Laços de Repetição:
  - ▶ **Laços Contados:** quando se conhece previamente quantas vezes será executado.
  - ▶ **Laços Condicionais:** quando não se conhece previamente quantas vezes será executado. Está vinculado a uma *condição*.



# Loop Enquanto

- ▶ A condição é testada no início.
- ▶ Se for VERDADEIRO o processo é executado e retorna para a condição inicial.
- ▶ Só sairá do processo quando a condição for FALSA.

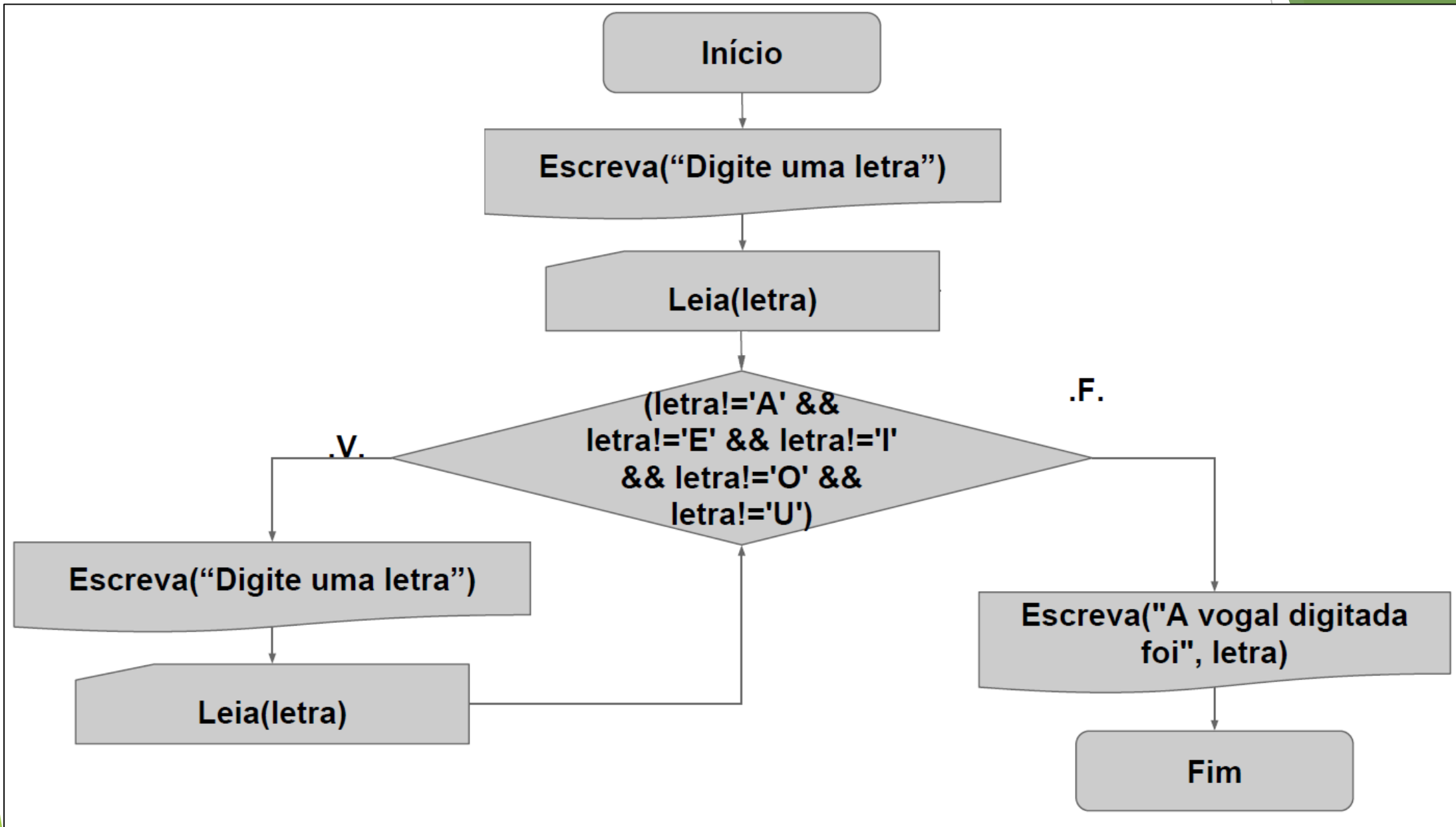


# Exemplo

- Faça um fluxograma que receba várias letras, quando for digitado uma vogal, o algoritmo deve interromper o processo e imprimir a mensagem “A vogal digitada foi XXX”.



# Exemplo





# Loop Enquanto

A construção "enquanto" (também chamada "repetição pré-testada") é a mais difundida estrutura de repetição, e sua estrutura básica é a seguinte:

```
Enquanto <condição> Faça  
    <bloco_de_código>  
Fim_enquanto
```

**<condição>**: expressão com resultado verdadeiro ou falso.

**<bloco\_de\_código>**: trecho de *código* executado quando a condição é verdadeira.



# Loop Enquanto

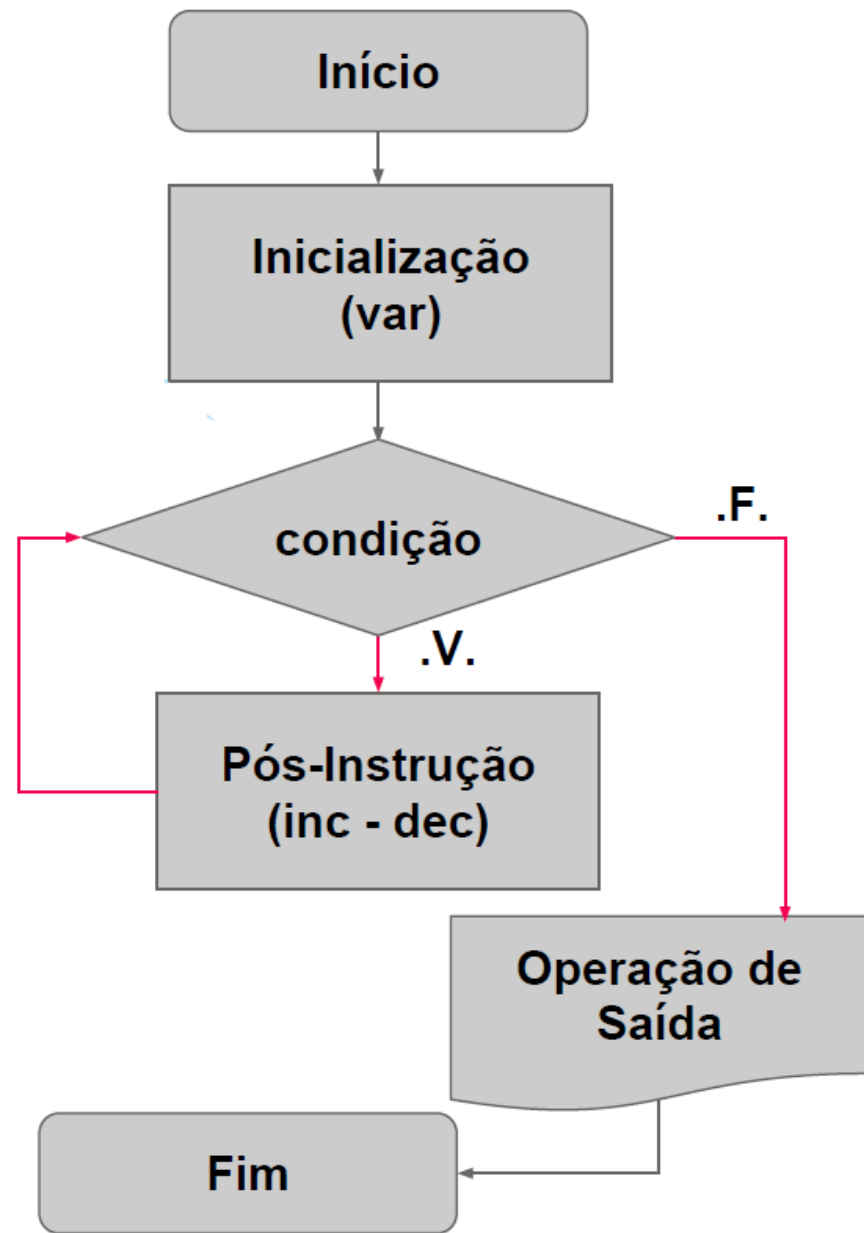
**Exemplo:** Ler um número da entrada, *enquanto* ele for diferente de zero.

```
Algoritmo lendo_enquanto_nao_zero
    var x:inteiro
Inicio
    Escreva("Digite um número diferente de 0")
    Leia(x)
    Enquanto x != 0 Faça
        Escreva("Digite um número diferente de 0")
        Leia(x)
    Fim_enquanto
    Escreva("Zero foi digitado")
Fim
```



# Loop Para

- ▶ Conhece previamente o número de vezes que será executado.
- ▶ A condição é testada no **início** do processo.
- ▶ **var** é necessariamente uma variável, o valor é **alterado** em cada interação.
- ▶ **inc** é o valor adicionado à variável var ao final de cada interação (incremento).
- ▶ **dec** é o valor subtraído à variável var ao final de cada interação (decremento).
- ▶ Só sairá do processo quando variável var for maior(inc) ou menor(dec) que o valor fim.



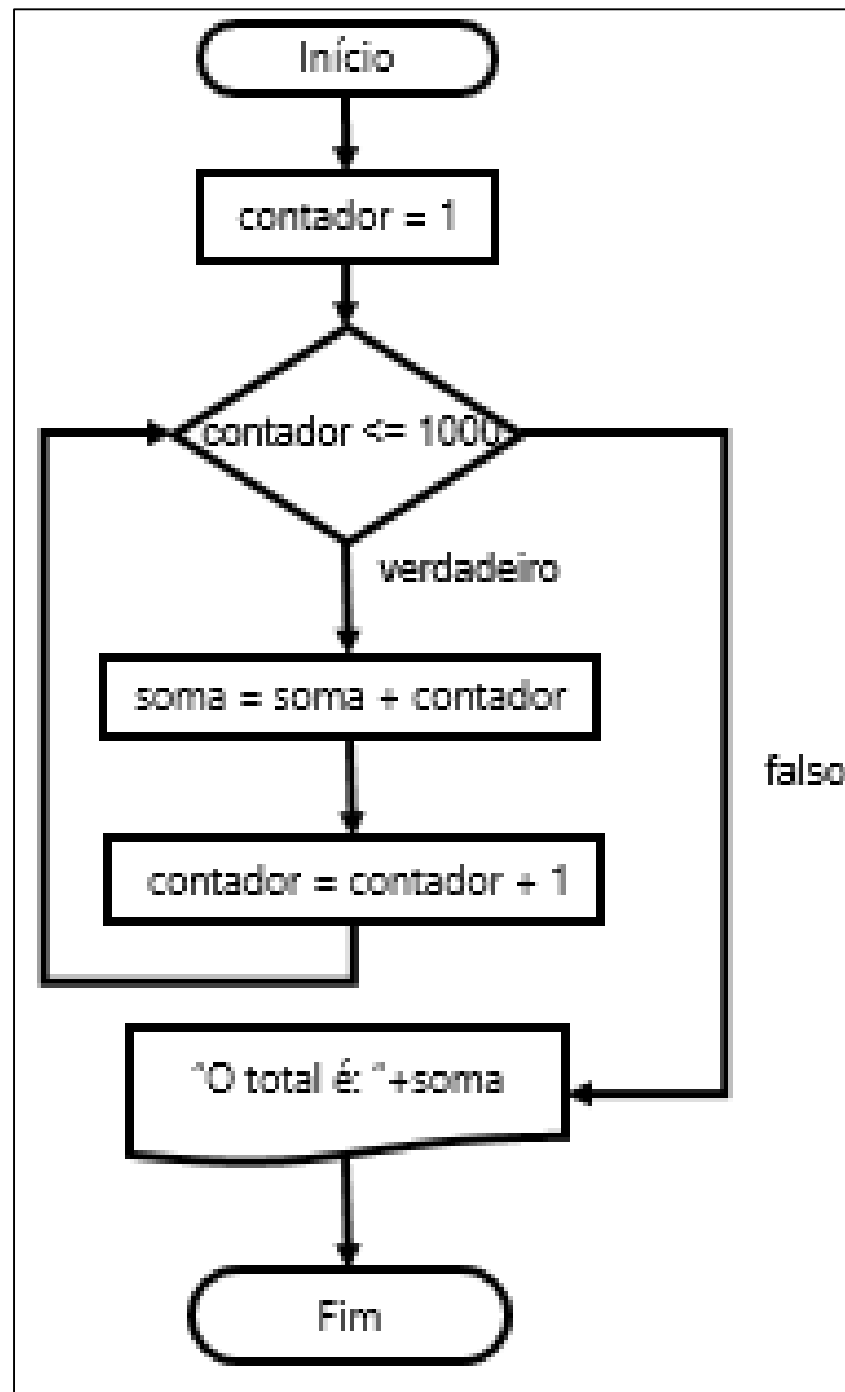
# Exemplo

- ▶ Faça uma fluxograma que some de 1 a 1000.



nemo

# Exemplo



# Loop Para

A construção "para" (ou "repetição com variável de controle") é uma estrutura de repetição que designa uma variável de controle para cada iteração do bloco, e uma operação de passo a cada iteração. Sua estrutura básica é a seguinte:

**Para** <v> **de** <vi> **até** <vf> **Passo** <p> **Faça**  
    <bloco\_de\_código>  
**Fim\_para**

<v>: variável contadora.

<vi>: valor *inicial* da variável de parada.

<vf>: valor *final* da variável de parada.

<p>: valor a ser incrementada.

<bloco\_de\_código>: trecho de código executado a cada iteração.



# Loop Para

**Exemplo:** Faça uma algoritmo que some de 1 a 1000.

```
Algoritmo soma_1_ate_1000
    var soma, contador:inteiro
Inicio
    Para contador de 1 até 1000 Passo 1
Faça
        soma = soma + contador
    Fim_para
    Escreva("O total é: "+soma)
Fim
```



# Considerações Finais

**<condição>**: expressão com resultado verdadeiro ou falso.

**<bloco\_de\_código>**: trecho de *código* executado quando a condição é verdadeira.

**Enquanto <condição> Faça**  
    <bloco\_de\_código>  
**Fim\_enquanto**

**<v>**: variável contadora.

**<vi>**: valor *inicial* da variável de parada.

**<vf>**: valor *final* da variável de parada.

**<p>**: valor a ser incrementada.

**<bloco\_de\_código>**: trecho de código executado a cada iteração.

**Para <v> de <vi> até <vf> Passo <p> Faça**  
    <bloco\_de\_código>  
**Fim\_para**



# Exercícios - Loops

1. Faça um pseudocódigo que receba dois números (***a*** e ***b***) e some todos os números **inteiros** de ***a*** a ***b***, e mostre o resultado obtido.
2. Faça um pseudocódigo e some todos os números **pares** e **inteiros** de **1** a **100**, e mostre o resultado obtido.
3. Faça um pseudocódigo que **some** os números digitados pelo usuário, até que seja informado o número -1.
4. Faça um pseudocódigo que exiba a tabuada de um número ***n***, de ***n*\*1** a ***n*\*10**.
5. Faça um pseudocódigo que leia uma quantidade indeterminada de **números inteiros** e **positivos** e indique qual foi o **menor** e **maior** número digitado, até que seja informado o número -1.
6. Faça um pseudocódigo que **calcule a média** de uma quantidade indeterminada de **números pares positivos**, até que seja informado o número 0.

# Exercícios - Loops

7. Faça um pseudocódigo que receba dois números (**a** e **b**). Verifique se esses números são maiores que 0 e, somente se forem, calcule: a soma todos os números **pares** de **a** a **b** a soma de todos os números **ímpares** de **a** a **b**. Mostre o resultado obtido.
  8. Faça um pseudocódigo que leia um número **n** e diga se ele é primo ou não.
  9. Faça um pseudocódigo que receba do usuário uma quantidade indeterminada de notas de provas e seus pesos. No fim calcule a média ponderada dessas notas (ex.: $\{[nota1*peso1]+[nota2*peso2]+\dots+[notan*peson]\}/soma\_dos\_pesos$ ). Leia do usuário enquanto não for digitado -1 para nota e -1 para o peso.
- Exemplo de entrada:
- nota = 7.4, peso = 2
  - nota = 8.6, peso = 3
  - nota = 10.0, peso = 5
  - nota = -1, peso = -1
- Saída desejada:
- "A média foi: 7,8"

# Exercícios - Loops

**10.** Faça uma pseudocódigo que leia um número N e calcule e imprima o seu fatorial.  
Ex:  $5! = 120$

**11.** Faça uma pseudocódigo que calcule a série de Fibonacci. No programa principal leia um número N e imprima o valor da sequencia na posição N. Serie de Fibonacci = 1, 1, 2, 3, 5, 8, 13, 21, 34...

**12.** Sendo  $h = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/N$ , construa um pseudocódigo para calcular o número h, sendo o número inteiro N fornecido pelo usuário. Seu algoritmo deve garantir que apenas um valor maior do que zero seja aceito como entrada, ou seja, somente calcular h se o n informado for maior do que 0, caso contrário, deve ser solicitado outro n, até que n seja maior do que 0. Ex.:  $n = 5$ ,  $h = 1 + 1/2 + 1/3 + 1/4 + 1/5$ .

**13.** Construa um pseudocódigo para determinar e mostrar o número de dígitos de um número inteiro informado. Ex.:  $n = 134859$ , imprimir 6;  $n = 111111111$ , imprimir 9.

# That's all Folks!



nemo