

Algoritmos - 3

Alexandre Diehl

Departamento de Física - UFPel



Estruturas básicas

Estruturas usadas num algoritmo

- ✓ Estrutura sequencial
- ✓ Estrutura condicional
- ✓ Estrutura de repetição

Estruturas básicas

Estruturas usadas num algoritmo

✓ Estrutura de repetição

Permite que uma sequência de **ações** ao longo do algoritmo seja **executada repetidamente, até** que uma determinada **condição de interrupção seja satisfeita**.

A **condição de interrupção** é representada por uma **expressão lógica**.

Tipos de estruturas de repetição

- **para**
- **enquanto**
- **repita**

Estruturas básicas

Repetição do tipo para

Usada quando **sabemos o número de repetições** a serem executadas na estrutura de repetição.

Exige o **uso de um identificador inteiro**, responsável pela contagem do número de repetições da estrutura.

Forma de uso

```
para indice <- valor_inicial ate valor_final faca passo n  
comando
```

indice: identificador inteiro de contagem de repetições. Deve ser declarado.

Estruturas básicas

Repetição do tipo para

Usada quando **sabemos o número de repetições** a serem executadas na estrutura de repetição.

Exige o **uso de um identificador inteiro**, responsável pela contagem do número de repetições da estrutura.

Forma de uso

```
para indice <- valor_inicial ate valor_final faca passo n  
comando
```

valor_inicial: valor inicial do identificador inteiro de contagem.

Estruturas básicas

Repetição do tipo para

Usada quando **sabemos o número de repetições** a serem executadas na estrutura de repetição.

Exige o **uso de um identificador inteiro**, responsável pela contagem do número de repetições da estrutura.

Forma de uso

```
para indice <- valor_inicial ate valor_final faca passo n  
comando
```

valor_final: valor final do identificador inteiro de contagem.

Estruturas básicas

Repetição do tipo para

Usada quando **sabemos o número de repetições** a serem executadas na estrutura de repetição.

Exige o **uso de um identificador inteiro**, responsável pela contagem do número de repetições da estrutura.

Forma de uso

```
para indice <- valor_inicial ate valor_final faca passo n  
comando
```

passo: incremento (ou decremento) **n** do identificador inteiro de contagem.

Estruturas básicas

Repetição do tipo para

Usada quando **sabemos o número de repetições** a serem executadas na estrutura de repetição.

Exige o **uso de um identificador inteiro**, responsável pela contagem do número de repetições da estrutura.

Forma de uso

```
para indice <- valor_inicial ate valor_final faca passo n  
comando
```

passo: se `valor_inicial < valor_final` temos um incremento. Neste caso o **passo n** deve ser um **inteiro positivo**.

Estruturas básicas

Repetição do tipo para

Usada quando **sabemos o número de repetições** a serem executadas na estrutura de repetição.

Exige o **uso de um identificador inteiro**, responsável pela contagem do número de repetições da estrutura.

Forma de uso

```
para indice <- valor_inicial ate valor_final faca  
    comando
```

passo: se o **incremento n** for **igual a 1**, podemos **omitir** a palavra **passo**.

Estruturas básicas

Repetição do tipo para

Usada quando **sabemos o número de repetições** a serem executadas na estrutura de repetição.

Exige o **uso de um identificador inteiro**, responsável pela contagem do número de repetições da estrutura.

Forma de uso

```
para indice <- valor_inicial ate valor_final faca passo n  
comando
```

passo: qualquer **incremento n** maior do que **1** deve ser declarado.

Estruturas básicas

Repetição do tipo para

Usada quando **sabemos o número de repetições** a serem executadas na estrutura de repetição.

Exige o **uso de um identificador inteiro**, responsável pela contagem do número de repetições da estrutura.

Forma de uso

```
para indice <- valor_inicial ate valor_final faca passo -n  
comando
```

passo: se $\text{valor_inicial} > \text{valor_final}$ temos um **decremento**. Neste caso o **passo** deve ser um **inteiro negativo -n**, que deve sempre ser declarado.

Estruturas básicas

Repetição do tipo para

Usada quando **sabemos o número de repetições** a serem executadas na estrutura de repetição.

Exige o **uso de um identificador inteiro**, responsável pela contagem do número de repetições da estrutura.

Forma de uso

```
para indice <- valor_inicial ate valor_final faca  
    comando
```

comando: ação que será executada caso **indice** esteja entre **valor_inicial** e **valor_final**. Com este formato, **apenas uma ação** será executada.

Estruturas básicas

Repetição do tipo para

Usada quando **sabemos o número de repetições** a serem executadas na estrutura de repetição.

Exige o **uso de um identificador inteiro**, responsável pela contagem do número de repetições da estrutura.

Forma de uso

```
para indice <- valor_inicial ate valor_final faca
  inicio
    comando1
    comando2
    comandosn
  fim
```

Forma para que **mais de uma ação** seja executada.

Estruturas básicas

Repetição do tipo para

Podemos usar **identificadores declarados** como os valores iniciais e finais do índice de contagem de repetições.

- O **passo** deve ser **sempre um número inteiro** positivo (incremento) ou negativo (decremento) declarado: não podem ser usados identificadores.

Forma de uso

algoritmo

```
declare a, b, j numero
a <- 1
b <- 100
para j <- a ate b faca passo 1
    comando
```

fim_algoritmo

Estruturas básicas

Repetição do tipo para

Podemos usar **identificadores declarados** como os valores iniciais e finais do índice de contagem de repetições.

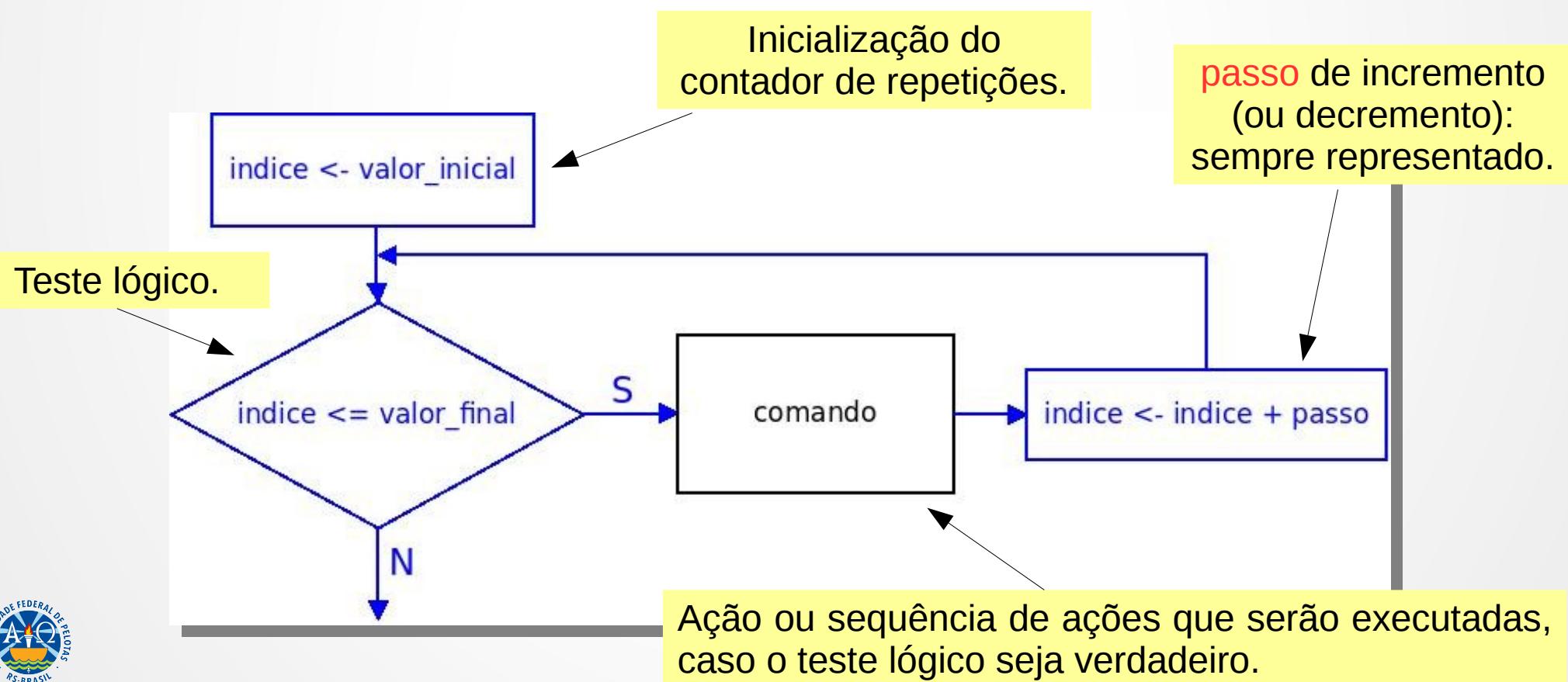
- A **mudança nos valores** destes identificadores **não terá impacto** no número de repetições da estrutura.

```
algoritmo
    declare a, b, j numero
    a <- 1
    b <- 100
    para j <- a ate b faca passo 1
    inicio
        a <- 2
        b <- 8
        escreva j
    fim
    escreva a, " ", b
fim_algoritmo
```

Estruturas básicas

Repetição do tipo para

Forma de representação num fluxograma



Estruturas básicas

Repetição do tipo para

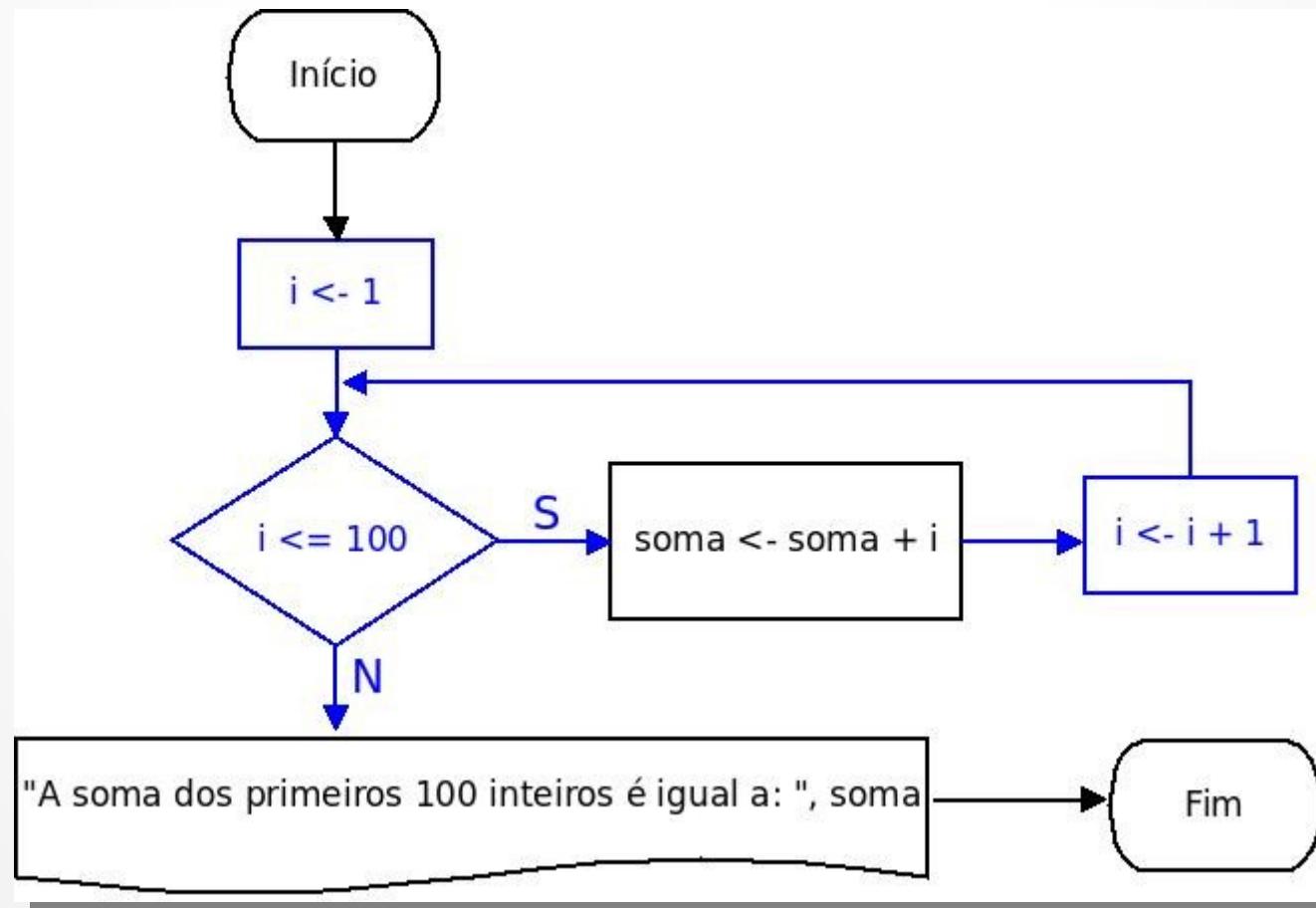
Exemplo 1: Algoritmo para somar os primeiros 100 números inteiros.

```
algoritmo
    declare i, soma numerico
    soma <- 0
    para i <- 1 ate 100 faca passo 1
        soma <- soma + i
    escreva "A soma dos primeiros 100 inteiros é igual a: ",soma
fim_algoritmo
```

Estruturas básicas

Repetição do tipo para

Exemplo 1: Algoritmo para somar os primeiros 100 números inteiros.



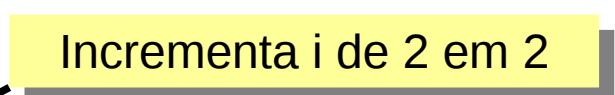
Estruturas básicas

Repetição do tipo para

Exemplo 2: Algoritmo para somar os **inteiros ímpares** de 1 até 100.

```
algoritmo
    declare i, soma numerico
    soma <- 0
    para i <- 1 ate 100 faca passo 2
        inicio
            soma <- soma + i
        fim
    escreva "A soma dos ímpares de 1 até 100 é igual a: ",soma
fim_algoritmo
```

Incrementa i de 2 em 2



Estruturas básicas

Repetição do tipo para

Exemplo 3: O valor aproximado de π pode ser calculado usando a série abaixo,

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$

Construa um algoritmo (pseudocódigo e fluxograma) para o cálculo de π , usando os primeiros N termos da soma acima (N deve ser fornecido pelo usuário).

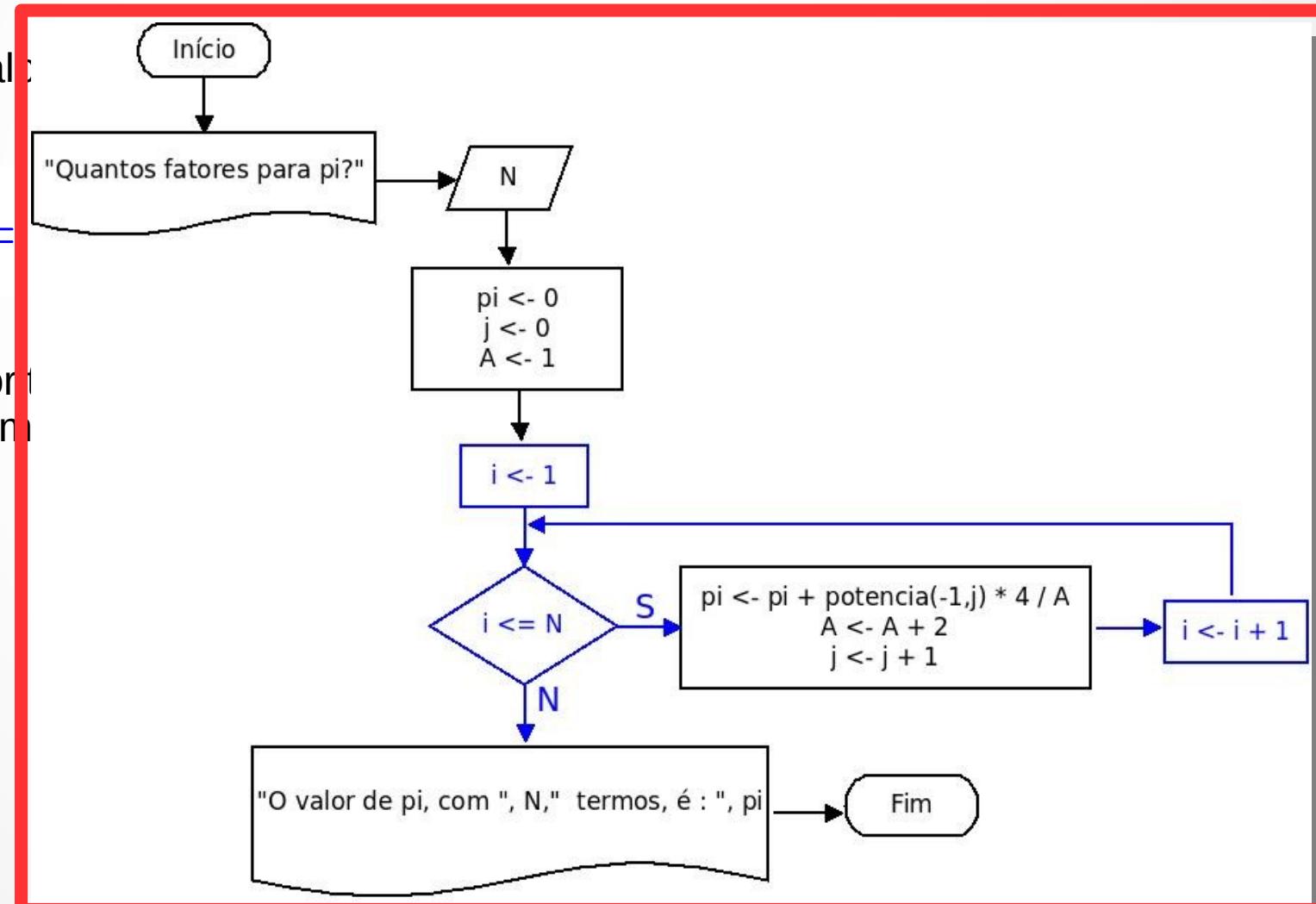
Estruturas básicas

Repetição do tipo para

Exemplo 3: O valor de π com N termos

Construa um algoritmo que calcule o valor de π com os primeiros N termos.

$\pi =$



Estruturas básicas

Repetição do tipo para

Exemplo 3: O valor aproximado de π

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$

algoritmo

```
declare N, pi, i, j, A numerico
escreva "Quantos fatores para pi?"
leia N
pi <- 0
j <- 0
A <- 1
para i <- 1 ate N faca
  inicio
    pi <- pi + potencia(-1, j)*4/A
    A <- A + 2
    j <- j + 1
  fim
escreva "O valor de pi, com ",N," termos, é : ",pi
fim_algoritmo.
```

Estruturas básicas

Repetição do tipo para

Exemplo 4: Faça um algoritmo (pseudocódigo) que calcula o resultado da soma abaixo,

$$\frac{2^1}{50} + \frac{2^2}{49} + \frac{2^3}{48} + \dots + \frac{2^{50}}{1}$$

Estruturas básicas

Repetição do tipo para

Exemplo 4: Faça um algoritmo (pseudocódigo) que calcula o resultado da soma abaixo,

```
algoritmo
    declare soma, denominador, i numérico
    soma <- 0
    denominador <- 50
    para i <- 1 ate 50 faca
        inicio
            soma <- soma + potencia(2, i) / denominador
            denominador <- denominador - 1
        fim
        escreva "O resultado da soma é : ", soma
    fim_algoritmo
```

Estruturas básicas

Repetição do tipo para

Exemplo 5: Faça um algoritmo que, a partir de um dado numérico N fornecido pelo usuário, calcule o fatorial de N .

$$N! = N \times (N - 1) \times (N - 2) \times (N - 3) \times \dots \times 3 \times 2 \times 1 \quad \text{para } N \geq 2$$

Estruturas básicas

Repetição do tipo para

Exemplo 5: Faça um algoritmo que, a partir de um dado numérico *N* fornecido pelo usuário, calcule o fatorial de *N*.

```
algoritmo
    declare factorial, j, N numerico
    escreva "Digite o valor do inteiro N a ser fatorado:"
    leia N
    factorial <- 1
    para j <- N ate 1 faca passo -1
    inicio
        factorial <- factorial * j
    fim
    escreva "O fatorial de N = ",N," vale : ", factorial
fim_algoritmo
```

Estruturas básicas

Estruturas usadas num algoritmo

✓ Estrutura de repetição

Permite que uma sequência de **ações** ao longo do algoritmo seja **executada repetidamente, até** que uma determinada **condição de interrupção seja satisfeita**.

A **condição de interrupção** é representada por uma **expressão lógica**.

Tipos de estruturas de repetição

- **para**
- **enquanto**
- **repita**

Estruturas básicas

Repetição do tipo enquanto

Usada quando **não sabemos, a priori, o número de repetições** a serem executadas na estrutura de repetição.

- Para a entrada na estrutura, uma **condição lógica verdadeira** deve ser proposta.
- Uma vez na estrutura, as repetições serão executadas até que a **condição lógica de entrada torne-se falsa (condição de saída)**.

Forma de uso

```
enquanto condicao faça  
    comando
```

condicao: teste lógico de entrada na estrutura (deve ser verdadeiro).

Estruturas básicas

Repetição do tipo enquanto

Usada quando **não sabemos, a priori, o número de repetições** a serem executadas na estrutura de repetição (ou mesmo sabemos este número).

- O **teste lógico de entrada** pode conter **operadores aritméticos, relacionais e lógicos**.
- O **teste lógico de entrada** **não** pode conter **operadores de atribuição**.

Forma de uso

```
enquanto condicao faca  
    comando
```

condicao: teste lógico de entrada na estrutura (deve ser verdadeiro).

Estruturas básicas

Repetição do tipo enquanto

Usada quando **não sabemos, a priori, o número de repetições** a serem executadas na estrutura de repetição.

- Para a entrada na estrutura, uma **condição lógica verdadeira** deve ser proposta.
- Uma vez na estrutura, as repetições serão executadas até que a **condição lógica de entrada torne-se falsa (condição de saída)**.

Forma de uso

```
enquanto condicao faca  
    comando
```

comando: ação (**única**) que será executada, caso **condicao** seja verdadeiro.

Estruturas básicas

Repetição do tipo enquanto

Usada quando **não sabemos, a priori, o número de repetições** a serem executadas na estrutura de repetição.

- Para a entrada na estrutura, uma **condição lógica verdadeira** deve ser proposta.
- Uma vez na estrutura, as repetições serão executadas até que a **condição lógica de entrada torne-se falsa (condição de saída)**.

Forma de uso

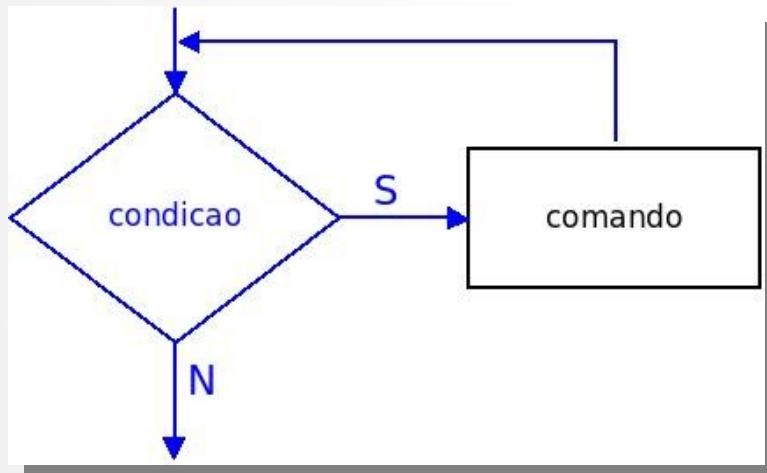
```
enquanto condicao faca
  inicio
    comando1
    comando2
  fim
```

comando: mais de uma **ação** será executada, caso **condicao** seja verdadeiro.

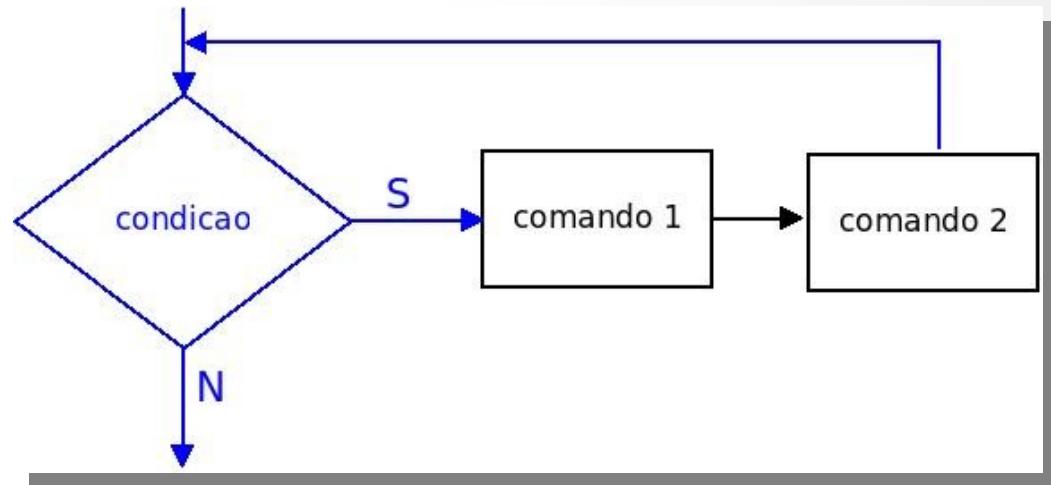
Estruturas básicas

Repetição do tipo enquanto

Forma de representação num fluxograma



Uma única ação será executada, caso condicao seja verdadeiro.



Mais de uma única ação será executada, caso condicao seja verdadeiro.

Estruturas básicas

Repetição do tipo enquanto

Exemplo 1: Algoritmo para somar os primeiros 100 números inteiros.

```
algoritmo
    declare j, soma numerico
    Acumulador da soma
    j <- 1
    soma <- 0
    enquanto j <= 100 faca
        inicio
            soma <- soma + j
            j <- j + 1
        fim
        escreva "A soma dos primeiros 100 inteiros é: ", soma
    fim_algoritmo
```

Condição de entrada:
operador relacional \leq

Comando 1

Comando 2

Estruturas básicas

Repetição do tipo enquanto

Exemplo 2: Algoritmo para exigir que um dado de entrada esteja compreendido entre 0 e 10. O usuário não deve ser informado de nada a respeito do dado.

```
algoritmo
    declare dado numerico
        flag logico

    flag <- falso
    enquanto (nao flag) faça
        inicio
            escreva "digite um número qualquer."
            leia dado
            se dado >= 0 e dado <= 10
                entao flag <- verdadeiro
            fim
            escreva "O dado digitado é : ", dado
        fim_algoritmo
```

Condição de entrada: negação da variável lógica **flag**

Estruturas básicas

Repetição do tipo enquanto

Exemplo 2: Algoritmo para exigir que um dado de entrada esteja compreendido entre 0 e 10.

- Modifique o algoritmo anterior, para que o mesmo interrompa o processamento após um dado número de tentativas, caso o usuário não digite o dado no intervalo pretendido.

Estruturas básicas

Estruturas usadas num algoritmo

✓ Estrutura de repetição

Permite que uma sequência de **ações** ao longo do algoritmo seja **executada repetidamente, até** que uma determinada **condição de interrupção seja satisfeita**.

A **condição de interrupção** é representada por uma **expressão lógica**.

Tipos de estruturas de repetição

- **para**
- **enquanto**
- **repita**

Estruturas básicas

Repetição do tipo repita

Usada quando **não sabemos, a priori, o número de repetições** a serem executadas na estrutura de repetição.

- Para a **entrada** na estrutura, **nenhuma condição é testada**.
- Uma vez na estrutura, as repetições serão executadas até que a **condição lógica** torne-se **verdadeira (condição de saída)**.

Forma de uso

```
repita  
    comando  
    ate condicao
```

```
repita  
    comando 1  
    comando 2  
    ate condicao
```

condicao: teste lógico de saída da estrutura (deve ser verdadeiro).

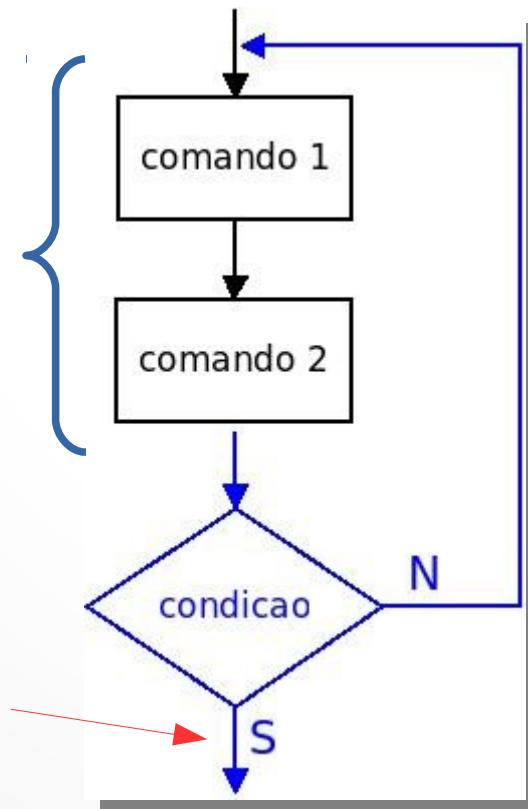
Estruturas básicas

Repetição do tipo repita

Forma de representação num fluxograma

Qualquer sequência
de ações dentro da
estrutura.

A condição deve ser
verdadeira para a saída
da estrutura



Estruturas básicas

Repetição do tipo repita

Exemplo 1: Algoritmo para somar os primeiros 100 números inteiros.

```
algoritmo
    declare j, soma numerico
    soma <- 0
    j <- 1
    repita
        soma <- soma + j
        j <- j + 1
    ate j > 100
    escreva "A soma dos primeiros 100 inteiros é :", soma
fim_algoritmo
```

soma <- 0 ← Acumulador da soma

Condição de saída:
operador relacional >

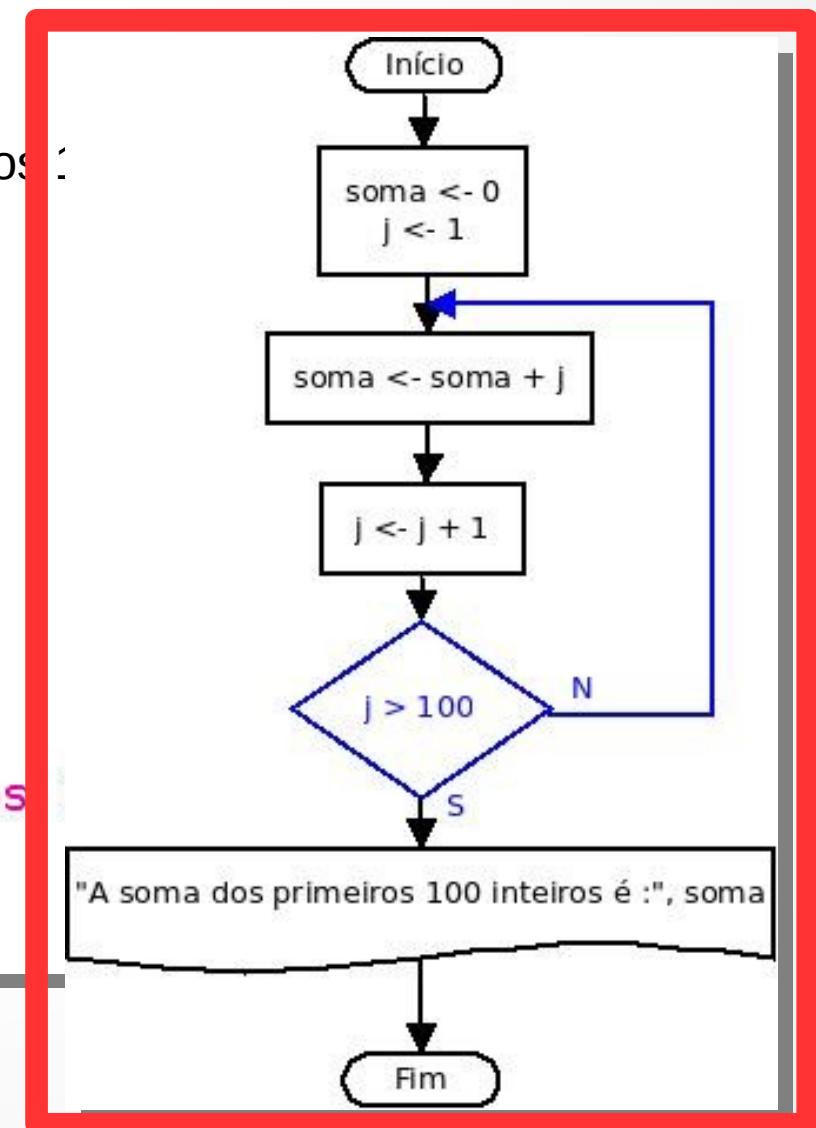
Estruturas básicas

Repetição do tipo repita

Exemplo 1: Algoritmo para somar os primeiros 100 inteiros

```
algoritmo
    declare j, soma numerico

    soma <- 0
    j <- 1
    repita
        soma <- soma + j
        j <- j + 1
    ate j > 100
    escreva "A soma dos primeiros"
    fim_algoritmo
```



Estruturas básicas

Repetição do tipo repita

Exemplo 2: Faça um algoritmo que, a partir de um dado numérico N fornecido pelo usuário, calcule o fatorial de N .

$$N! = N \times (N - 1) \times (N - 2) \times (N - 3) \times \dots \times 3 \times 2 \times 1 \quad \text{para } N \geq 2$$

- (a) use uma estrutura de repetição do tipo **ENQUANTO**.
- (b) use uma estrutura de repetição do tipo **REPITA**.