

Estruturas de repetição / laços condicionais

São estruturas de controle que permitem que um determinado trecho do programa/ algoritmo seja executado repetidamente, a partir da análise de uma condição, e que será interrompido/finalizado também a partir da análise da mesma condição.

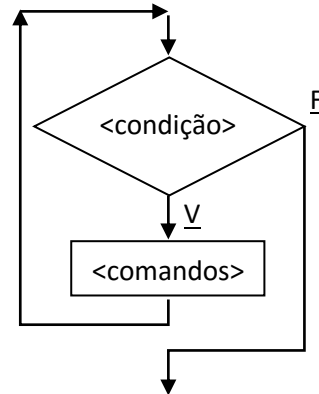
1 – com verificação da condição no início da estrutura/laço

Executa o(s) comando(s) associado(s) à estrutura 0(zero) ou n vezes, enquanto a condição for verdadeira.

Sintaxe:

```
enquanto <condição> faça
    <comando(s)>
fimenquanto
```

Exemplo:



```
algoritmo "exemplo_enquanto"
//mostra na tela a tabuada de um número informado pelo usuário
var
    n, x: inteiro
inicio
    escreval("Mostra na tela a tabuada de um número informado pelo usuário")
    escreva("Digite um número inteiro: ")
    leia(n)
    x <- 1
    enquanto x <= 10 faça
        escreval(n, " x ", x, " = ", n*x)
        x <- x + 1
    fimenquanto
fimalgoritmo
```

Neste ponto a variável de controle do laço de repetição é inicializada.

A condição do laço de repetição é uma expressão lógica.

A variável de controle é incrementada, caso contrário a condição de controle sempre será verdadeira e o laço nunca terminará (ficará em *loop infinito*).

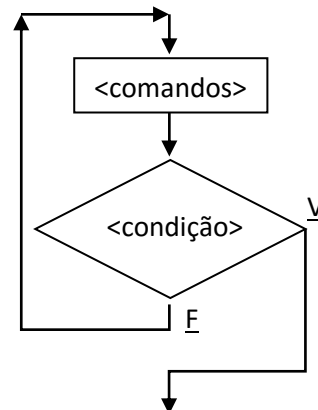
2 – com verificação da condição ao final da estrutura/laço

Executa o(s) comando(s) associado(s) à estrutura 1(uma) ou n vezes, enquanto a condição for verdadeira.

Sintaxe:

```
repita  
    <comando(s)>  
até <condição>
```

Exemplo:



```
algoritmo "exemplo_repita"  
//mostra na tela a tabuada de um número informado pelo usuário  
var  
    n, x: inteiro  
inicio  
    escreval("Mostra na tela a tabuada de um número informado pelo usuário")  
    escreva("Digite um número inteiro: ")  
    leia(n)  
    x <- 1  
    repita  
        escreval(n, " x ", x, " = ", n*x)  
        x <- x + 1  
    até x > 10  
fimalgoritmo
```

Neste ponto a variável de controle do laço de repetição é inicializada.

A variável de controle é incrementada, caso contrário a condição de controle sempre será verdadeira e o laço nunca terminará (ficará em *loop infinito*).

A condição do laço de repetição é uma expressão lógica.

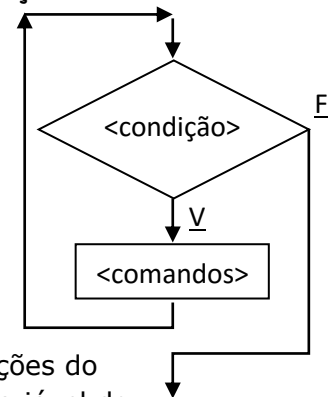
3 – com verificação no início e incrementação/decrementação da variável de controle implícita na estrutura

Executa o(s) comando(s) associado(s) à estrutura 1(uma) ou n vezes, até que a condição seja verdadeira.

Sintaxe:

```
para <variável> de <valor inicial> ate <condição>
[passo <incremento>] faca
    <comando(s)>
fimpara
```

onde:



<variável> É a variável contadora que controla o número de repetições do laço. Na versão atual, deve ser necessariamente uma variável do tipo inteiro, como todas as expressões deste comando.

<valor-inicial> É uma expressão que especifica o valor de inicialização da variável contadora antes da primeira repetição do laço.

<valor-limite> É uma expressão que especifica o valor máximo que a variável contadora pode alcançar.

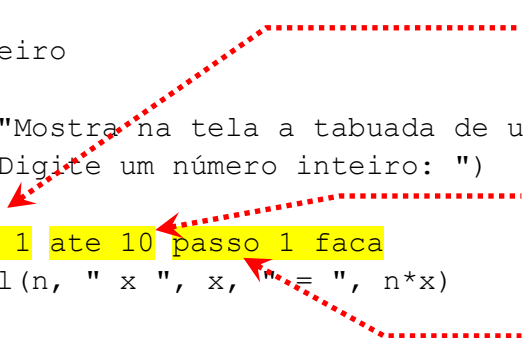
<incremento> É opcional. Quando presente, precedida pela palavra passo, é uma expressão que especifica o incremento que será acrescentado à variável contadora em cada repetição do laço. Quando esta opção não é utilizada, o valor padrão de <incremento> é 1. Vale a pena ter em conta que também é possível especificar valores negativos para <incremento>. Por outro lado, se a avaliação da expressão <incremento> resultar em valor nulo, a execução do algoritmo será interrompida, com a impressão de uma mensagem de erro.

fimpara Indica o fim da seqüência de comandos a serem repetidos. Cada vez que o programa chega neste ponto, é acrescentado à variável contadora o valor de <incremento>, e comparado a <valor-limite>. Se for menor ou igual (ou maior ou igual, quando <incremento> for negativo), a seqüência de comandos será executada mais uma vez; caso contrário, a execução prosseguirá a partir do primeiro comando que esteja após o fimpara.

<valor-inicial>, <valor-limite> e <incremento> são avaliados uma única vez antes da execução da primeira repetição, e não se alteram durante a execução do laço, mesmo que variáveis eventualmente presentes nessas expressões tenham seus valores alterados.

Exemplo:

```
algoritmo "exemplo_para"
//mostra na tela a tabuada de um número informado pelo usuário
var
    n, x: inteiro
inicio
    escreval("Mostra na tela a tabuada de um número informado pelo usuário")
    escreva("Digite um número inteiro: ")
    leia(n)
    para x de 1 ate 10 passo 1 faca
        escreval(n, " x ", x, " = ", n*x)
    fimpara
fimalgoritmo
```



Neste ponto a variável de controle do laço de repetição é inicializada.

A condição do laço de repetição é uma expressão lógica.

A variável de controle é incrementada na própria estrutura.

Condições de parada para estruturas de repetição.

1 – CONTADOR

Um **contador** é utilizado quando se sabe, com antecedência:

- quantas vezes a estrutura de repetição será executada
- quando se deseja saber quantas vezes ele foi executada

Exemplo:

```
algoritmo "exemplo_enquanto"
//Para um grupo de 50 pessoas, encontrar:
//a) a média de idade dos homens;
//b) a mulher mais nova.
var
  x, idade, sexo, somaidade, qhomens, mmn: real
inicio
  escreval("Para um grupo de 50 pessoas, encontrar:")
  escreval("a) a média de idade dos homens;")
  escreval("b) a mulher mais nova.")
  somaidade <- 0
  qhomens <- 0
  mmn <- 1000
  x <- 1
  enquanto x <= 50 faca
    escreva("Informe a idade da ", x+1, "ª pessoa: ")
    leia(idade)
    escreva("Digite 1 para masculino ou 2 para feminino: ")
    leia(sexo)
    se (sexo = 1) entao
      somaidade <- somaidade + idade
      qhomens <- qhomens + 1
    senao
      se mmn > idade entao
        mmn <- idade
      fimse
    fimse
    x <- x + 1
  fimenquanto
  escreval("A média de idade dos homens é :", (somaidade/qhomens):2:1)
  escreval("A mulher mais nova tem ", mmn, " anos")
fimalgoritmo
```

As variáveis que conterão os dados a serem encontrados no programa são inicializadas.

Neste ponto a variável de controle do laço de repetição é inicializada.

Aqui a idade dos homens é somada, além de ser contada sua quantidade

Aqui é realizada a verificação para que a idade da mulher mais nova seja armazenada na variável **mmn**

A condição do laço de repetição é uma expressão lógica.

A variável de controle é incrementada, caso contrário a condição de controle sempre será verdadeira e o laço nunca terminará (ficará em *loop infinito*).

2 – ACUMULADOR

Um **acumulador** é utilizado quando:

- não sabemos quantas vezes a estrutura de repetição será inicializada e o valor da variável de controle é definido dentro da própria estrutura sem a interferência do usuário do programa/algoritmo.

Exemplo:

```
algoritmo "exemplo_enquanto"
//Um determinado material radioativo perde 50% de sua massa a cada 73
//segundos.
//Determinar em quanto tempo (em minutos), dada a massa do material em
//gramas, será necessário para que a mesma seja menor do que 0,05
var
    tempo, massa: real
inicio
    escreval("Um determinado material radioativo perde 50% de sua massa a cada
        73 segundos.")
    escreval("Determinar em quanto tempo (em minutos), dada a massa do material
        em gramas, será necessário para que a mesma")
    escreval("seja menor do que 0,05")
    escreva("Informe a massa do material: ")
    leia(massa)
    tempo <- 0
    enquanto massa >= 0.05 faca
        massa <- massa / 2
        tempo <- tempo + 73
    fimenquanto
    escreval("O tempo necessário é de ", (tempo/60):2:2, " minutos")
finalgoritmo
```

A variável que conterá o tempo é inicializada.

A condição do laço de repetição é uma expressão lógica.

A variável de controle é decrementada, caso contrário a condição de controle sempre será verdadeira e o laço nunca terminará (ficará em *loop infinito*).

O tempo é somado a cada alteração da variável de controle do laço de repetição

3 – FLAG

Um **flag** é utilizado quando:

- não sabemos quantas vezes a estrutura de repetição será inicializada e o valor da variável de controle é alterado por interferência direta do usuário do programa/ algoritmo, ou por alguma condição alcançada no mesmo.

Exemplo:

```
algoritmo "exemplo_enquanto"
//Para um grupo de 50 pessoas, encontrar:
//a) a média de idade dos homens;
//b) a mulher mais nova.
var
    x, idade, sexo, somaidade, qhomens, mmn: real
    fim: caracter
inicio
    escreval("Para um grupo de 50 pessoas, encontrar:")
    escreval("a) a média de idade dos homens;")
    escreval("b) a mulher mais nova.")
    somaidade <- 0
    qhomens <- 0
    mmn <- 1000
    fim <- "sim"
    enquanto fim = "sim" faca
        escreva("Informe a idade da ", x+1, "ª pessoa: ")
        leia(idade)
        escreva("Digite 1 para masculino ou 2 para feminino: ")
        leia(sexo)
        se (sexo = 1) entao
            somaidade <- somaidade + idade
            qhomens <- qhomens + 1
        senao
            se mmn > idade entao
                mmn <- idade
            fimse
        fimse
        escreval("Deseja continuar?")
        escreva("Digite SIM para continuar ou NÃO para encerrar: ")
        leia(fim)
    fimenquanto
    escreval("A média de idade dos homens é :", (somaidade/qhomens):2:1)
    escreval("A mulher mais nova tem ", mmn, " anos")
fimalgoritmo
```

As variáveis que conterão os dados a serem encontrados no programa são inicializadas.

Neste ponto a variável de controle do laço de repetição é inicializada.

Aqui a idade dos homens é somada, além de ser contada sua quantidade

Aqui é realizada a verificação para que a idade da mulher mais nova seja armazenada na variável **mmn**

A condição do laço de repetição é uma expressão lógica.

A variável de controle é **alterada**, caso contrário a condição de controle sempre será verdadeira e o laço nunca terminará (ficará em *loop infinito*).