



Algoritmos e Programação

Aula 05 – Estrutura de Repetição (enquanto e repita)



Objetivos de Aprendizagem

- Identificar na situação problema a existência de aplicar uma estrutura de repetição com teste no início;
- Desenvolver algoritmos com estruturas de repetição.



Levantamento de conhecimentos prévios

- Quizz para relembrar conceitos da aula anterior
- Quizz no início da aula com no máximo 5 perguntas.





Estruturas de Controle de Fluxo

- Em certos algoritmos percebemos que a execução de determinados trechos se faz necessária algumas vezes.
 - Um exemplo que nós já tratamos é o caso do cálculo da média aritmética entre alguns valores.
- Vimos que a operação de leitura é repetida de acordo com o número de valores que servirão de base para o cálculo da média.
- Outra forma de verificar a necessidade de repetição de um determinado trecho do algoritmo é imaginar que ao invés de se desejar apenas obter uma média, o objetivo fosse obter um conjunto de médias. Para o cálculo de cada uma das médias o mesmo conjunto de operações seria executado.
 - Observações como estas fomentaram a criação das estruturas de repetição as quais veremos a seguir.



Estrutura ou Laço de Repetição

- Conjunto de estruturas sintáticas que permitem que um trecho de um algoritmo seja repetido um determinado número de vezes, sem que o código correspondente tenha que ser escrito mais de uma vez.
- enquanto ... faça
- repita ate ...
- para ... faça



Laço de repetição: enquanto

- Sintaxe:

...

```
enquanto (<expressão lógica>) faça  
    <sequência de comandos>  
fimenquanto
```

...



Laço de repetição: enquanto

- As instruções contidas no enquanto serão executadas durante o tempo em que o resultado da avaliação da expressão lógica resultar verdadeiro.
- O fato da avaliação da expressão lógica encontrar-se no início do laço faz com que a sequência de comandos só venha a ser executada se ao menos uma vez a avaliação da expressão resultar em verdadeiro.





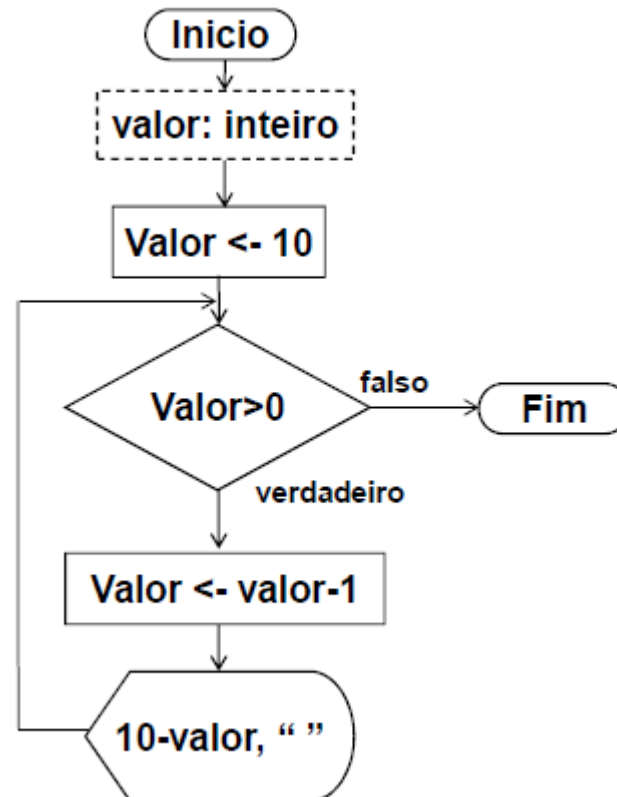
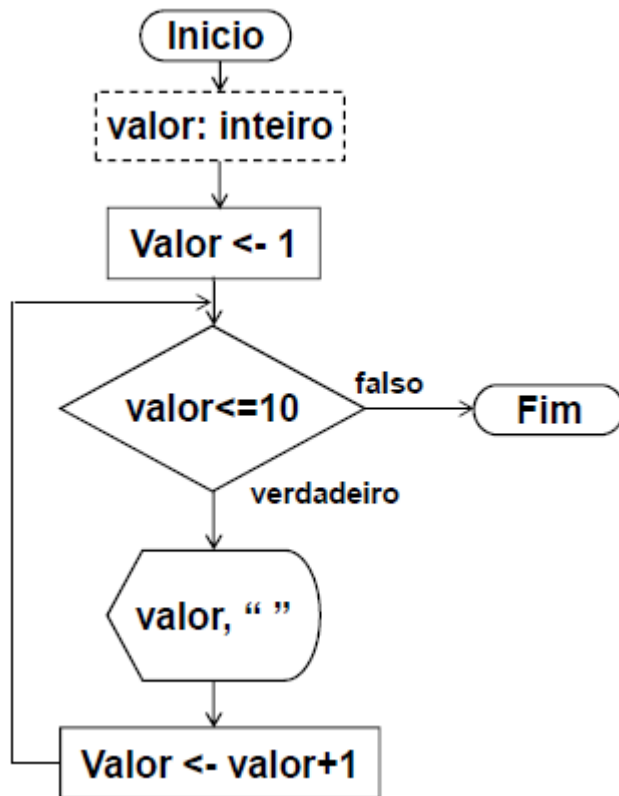
Exemplo

- O pseudocódigo e os fluxogramas a seguir representam algoritmos que escrevem na saída padrão os números inteiros contidos no intervalo $[1, 10]$.

Exemplo

```
algoritmo "exemplo 1 laço enquanto"
var
    valor: inteiro
inicio
    valor <- 1
    enquanto (valor <= 10) faca
        escreval (valor)
        valor <- valor+1
    fimenquanto
fimalgoritmo
```

Exemplo





Exemplo

- Dada uma reta $ax + by + c = 0$ e cinco pontos, faça um algoritmo para calcular, para cada ponto, o seguinte:
 - Se o ponto estiver no primeiro quadrante calcule e informe a distância do ponto a reta
 - Caso contrário escreva uma mensagem informando que o ponto não pertence ao primeiro quadrante.

Exemplo

```
algoritmo "exemplo 2 laço enquanto "  
Var  
    a,b,c,x,y: real  
    contador: inteiro  
inicio  
    contador <- 1  
    escreval ("Equação da reta:  $ax+by+c=0$ ")  
    escreva ("Coeficiente a da reta = ")  
    leia (a)  
    escreva ("Coeficiente b da reta = ")  
    leia (b)  
    escreva ("Coeficiente c da reta = ")  
    leia (c)
```

Exemplo

```
enquanto (contador<=5) faca
    escreval ("Coordenadas do ponto ",contador," :")
    escreva ("Coordenada x do ponto = ")
    leia (x)
    escreva ("Coordenada y do ponto = ")
    leia (y)
    se (x>=0) e (y>=0) entao
        escreval ("A distância do ponto ",contador, " a reta é:
", ((a*x+b*y+c)^2)^0.5/(((a^2)+(b^2))^0.5))
    senao
        escreval ("O ponto não está no primeiro
quadrante! ")
    fimse
    contador <- contador + 1
fimenquanto
finalgoritmo
```



Laço de repetição: repita

- Sintaxe:

...

repita

<sequência-de-comandos>

ate (<expressão-lógica>)

...



Laço de repetição: repita

- As instruções contidas no repita serão executadas enquanto o resultado da avaliação da expressão lógica resultar em falso.
- O fato da avaliação da expressão lógica encontrar-se no final do laço faz com que, mesmo no caso da expressão lógica nunca resultar em falso, a sequência de comandos seja executada ao menos uma vez.



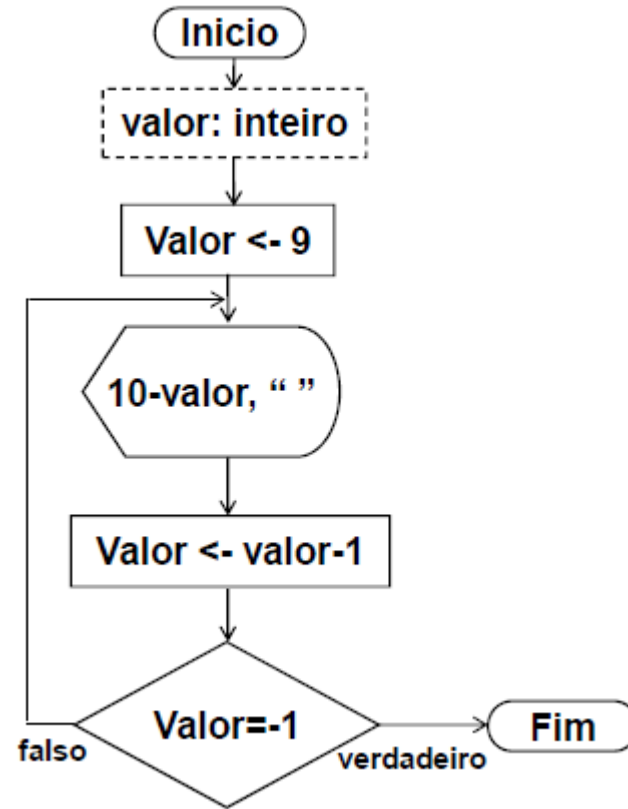
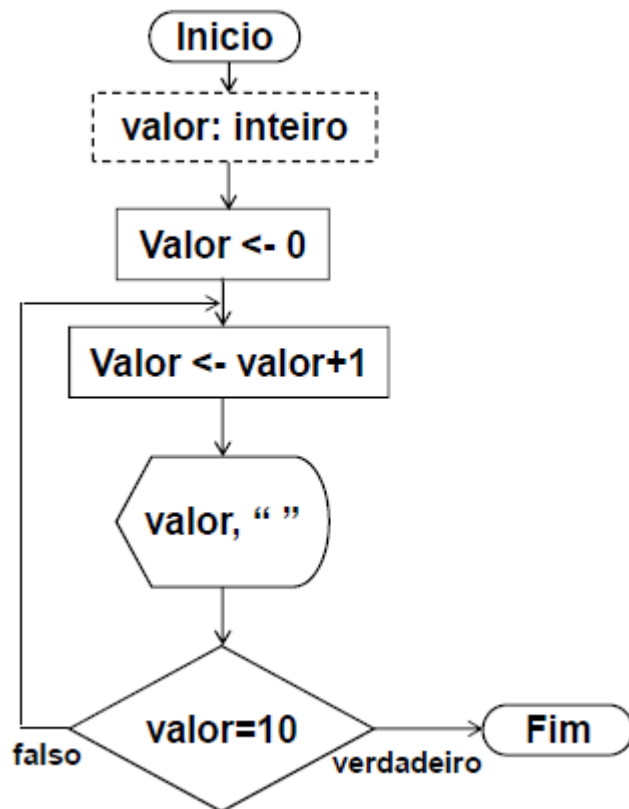
Exemplo

- O pseudocódigo e os fluxogramas a seguir representam algoritmos que escrevem na saída padrão os números inteiros contidos no intervalo $[1, 10]$.

Exemplo

```
algoritmo "exemplo 5"
var
    valor: inteiro
inicio
    valor <- 0
    repita
        valor <- valor+1
        escreval (valor)
    ate (valor = 10)
fimalgoritmo
```

Exemplo





Exemplo


- Dada uma reta $ax + by + c = 0$ e cinco pontos, faça um algoritmo para calcular, para cada ponto, o seguinte:
 - Se o ponto estiver no primeiro quadrante calcule e informe a distância do ponto a reta
 - Caso contrário escreva uma mensagem informando que o ponto não pertence ao primeiro quadrante.

Exemplo

```
algoritmo "exemplo 6"
Var
    a,b,c,x,y: real
    contador: inteiro
inicio
    contador <- 1
    escreval ("Equação da reta: ax+by+c=0")
    escreva ("Coeficiente a da reta = ")
    leia (a)
    escreva ("Coeficiente b da reta = ")
    leia (b)
    escreva ("Coeficiente c da reta = ")
    leia (c)
```

Exemplo

```
repita
  escreval ("Coordenadas do ponto ",contador," :")
  escreva ("Coordenada x do ponto = ")
  leia (x)
  escreva ("Coordenada y do ponto = ")
  leia (y)
  se ((x>=0) e (y>=0)) entao
    escreval ("A distância do ponto ",contador, " a reta é: ",
((a*x+b*y+c)^2)^0.5/(((a^2)+(b^2))^0.5))
  senao
    escreval ("O ponto ",contador," não está no primeiro
quadrante! ")
  fimse
  contador <- contador + 1
ate (contador>5)
finalgoritmo
```



Pare-Pense-Compartilhe

- Faça um algoritmo que recebe números naturais fornecidos pelo usuário, quando o usuário quiser parar a execução do algoritmo, o mesmo fornecerá um número negativo.
- O algoritmo deve retornar, ao final de seu processamento, a quantidade de números naturais fornecida pelo usuário.
- Fazer dois algoritmos utilizando em cada um, uma das estruturas de repetição vistas. Os algoritmos desenvolvidos devem ser representados através de um pseudocódigo e de um fluxograma.

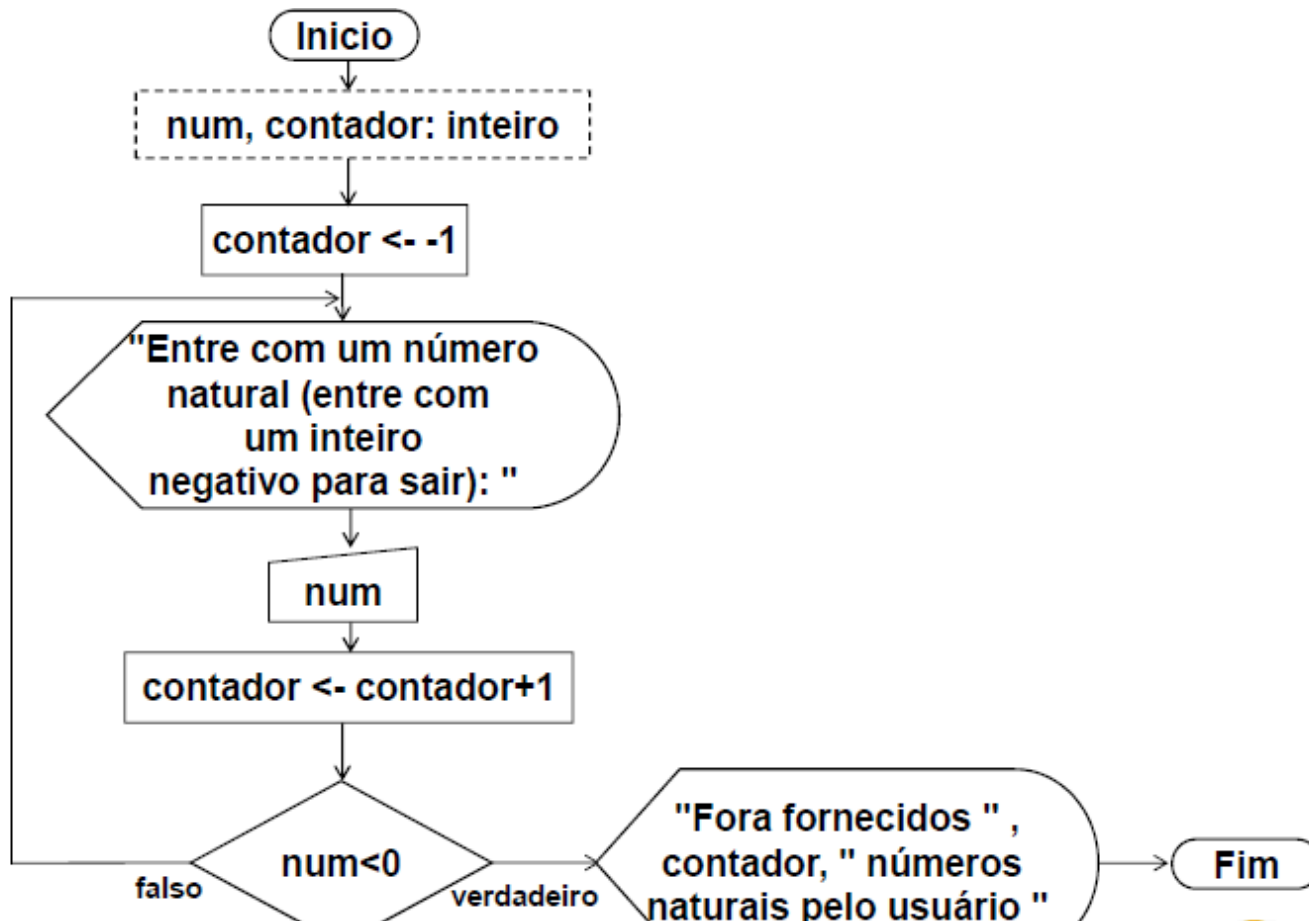
Resolução


```
algoritmo "exercício 15 laço de repetição repita a"
Var
    num, contador: inteiro
inicio
    contador <- 0
    repita
        escreva ("Entre com um número natural (entre com um inteiro negativo para sair): ")
        leia (num)
        se (num >= 0) entao
            contador <- contador + 1
        fimse
    ate (num < 0)
    escreva ("Fora fornecidos " , contador, " números naturais pelo usuário ")
finalgoritmo
```


Resolução

```
algoritmo " exercício 15 laço de repetição repita b"  
Var  
    num, contador: inteiro  
inicio  
    contador <- -1  
    repita  
        escreva ("Entre com um número natural (entre com  
um inteiro negativo para sair): ")  
        leia (num)  
        contador <- contador + 1  
    ate (num<0)  
    escreva ("Fora fornecidos " ,contador, " números  
naturais pelo usuário ")  
fimalgoritmo
```

Fluxograma





Pare-Pense-Compartilhe

- Com base no que foi exposto, construa um fluxograma para obter o resultado da divisão entre dois números.
- **Observação:** Caso um dos operandos não seja válido o mesmo deve ser novamente solicitado até um valor válido ser fornecido, ou seja, as entradas devem ser validadas.



Atividade de Fixação

- Lista de atividades disponível no BlackBoard/WhatsApp:
 - Lista de Fixação III



Atividade Extraclasse

- Capítulo 5 do livro Fundamentos da Programação de Computadores (ASCENCIO).
- Exercícios para casa: capítulo 5 do livro Fundamentos da Programação de Computadores (ASCENCIO).



One Minute Paper

- Escreva um parágrafo refletindo sobre o processo de aprendizagem ocorrido na aula.
 1. O que aprendi de mais importante nesta aula?
 2. Qual a maior dúvida que ficou?
 3. Com que colegas da classe eu discuti a última aula?
- Acesso ao Google Forms:
 - bit.ly/omp-algoritmos