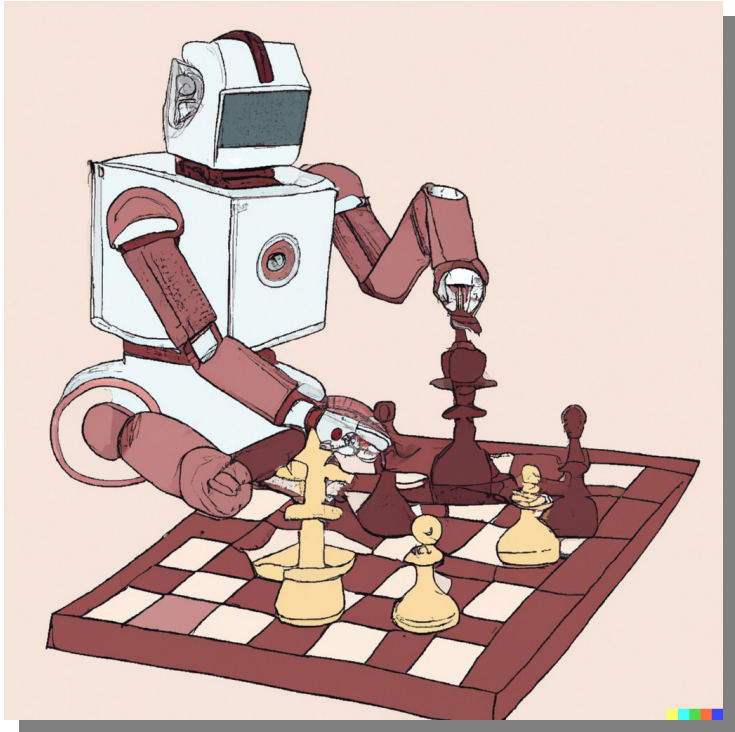




Universidade de Brasília

Departamento de Ciência da Computação



Semana 04

Algoritmos: Estruturas Algoritmicas

CIC0004

Algoritmos e Programação de Computadores

Prof. Pedro Garcia Freitas

<https://pedrogarcia.gitlab.io/>

pedro.garcia@unb.br

Brasilia



Este conjunto de slides não deve ser utilizado ou republicado sem a expressa permissão do autor.

This set of slides should not be used or republished without the author's express permission.



1. Objetivos

Esta aula introduz estende os conceitos para a representação de algoritmos, especialmente as estruturas algorítmicas.



2. Recapitulação

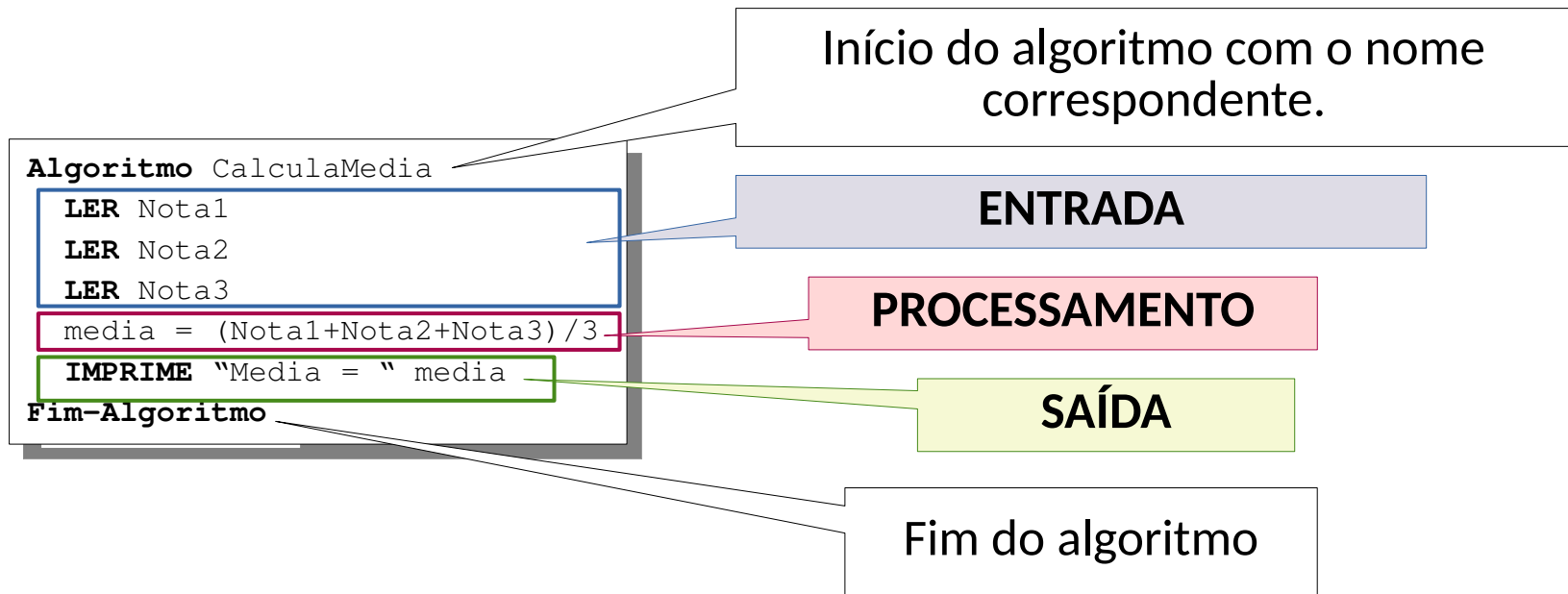
Última aula:

- Fases fundamentais de um algoritmo
 - Entrada, processamento e saída
- Formas de expressar um algoritmo
 - Pseudocódigo, fluxograma



2. Recapitulação

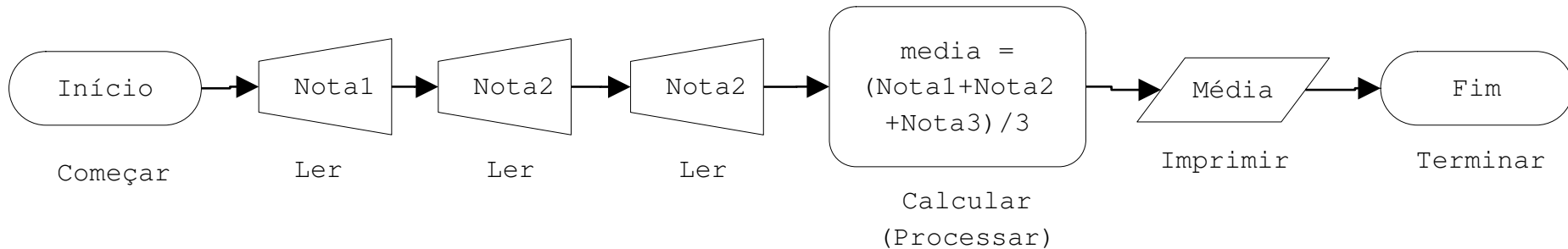
Última aula: sintaxe básica de um algoritmo sequencial (pseudocódigo)





2. Recapitulação

Última aula: sintaxe básica de um algoritmo sequencial (diagrama)





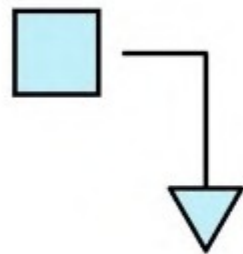
2. Recapitulação

- Algoritmo para construção de algoritmos.
- Basicamente a construção de um algoritmo se resume às seguintes etapas:

```
Algoritmo EscreverAlgoritmo  
  ENTENDER do problema;  
  DEFINIR os dados e respectivos tipos (variáveis);  
  OBTER esses dados (entradas);  
  ELABORAR processamento em si;  
  EXIBIR os resultados (saídas).  
Fim-Algoritmo
```

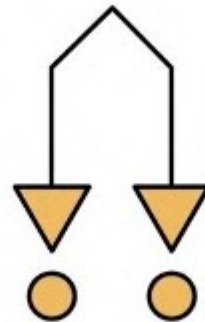
3. Estruturas Algorítmicas

- Nos algoritmos temos 3 estruturas básicas de controle. São elas:



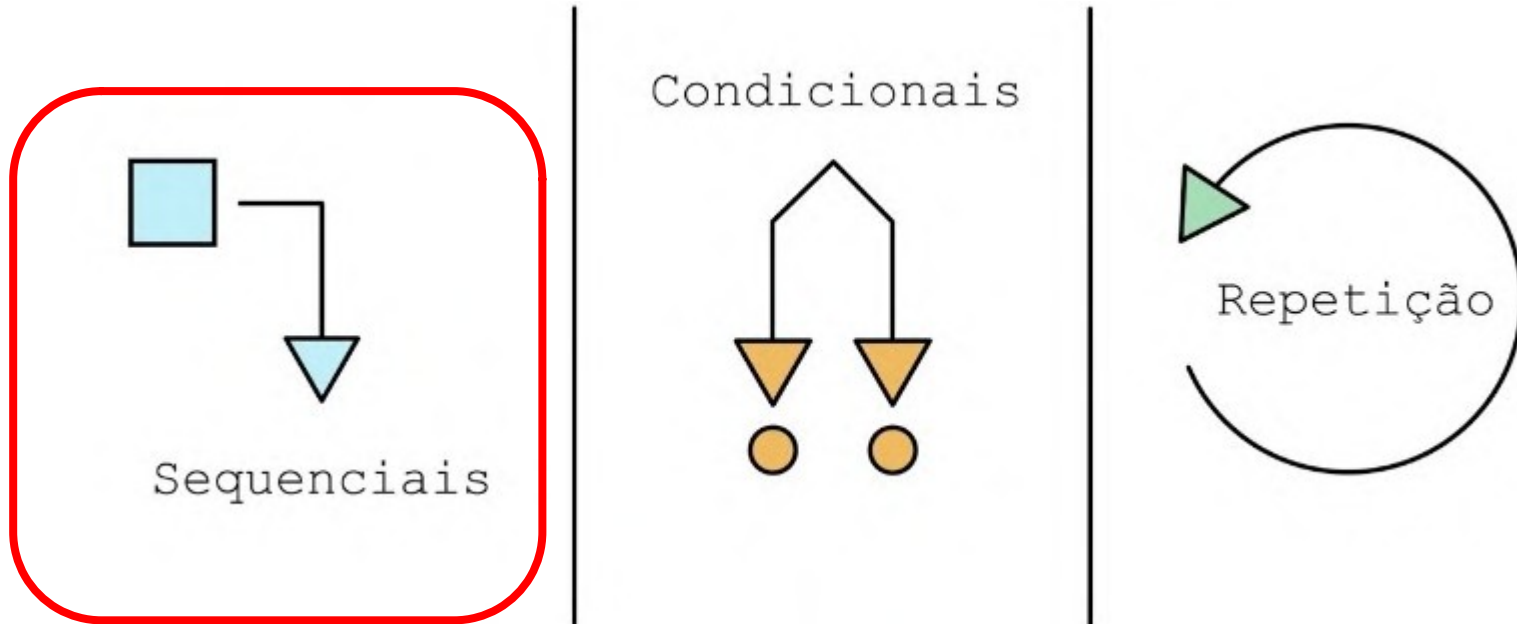
Sequenciais

Condicionais



3. Estruturas Algorítmicas

- Nos algoritmos temos 3 estruturas básicas de controle. São elas:



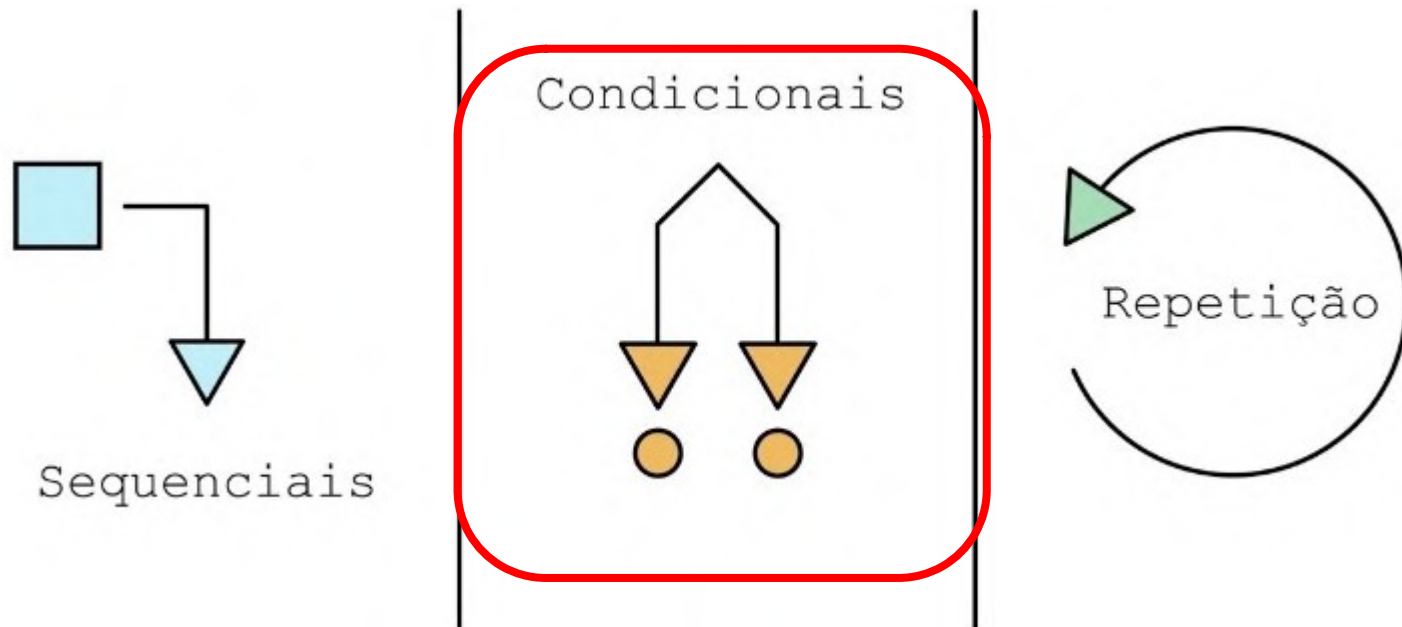
3. Estruturas Algorítmicas

- Estrutura sequencial é um conjunto de instruções no qual cada instrução será executada em sequência.
- São os exemplos que vimos até na última aula.



3. Estruturas Algorítmicas

- Nos algoritmos temos 3 estruturas básicas de controle. São elas:





3. Estruturas Algorítmicas

- As estruturas condicionais, também conhecidas como **estruturas seletivas**, são utilizadas para tomar **decisões lógicas**.



3. Estruturas Algorítmicas

- As estruturas condicionais, também conhecidas como **estruturas seletivas**, são utilizadas para tomar **decisões lógicas**.
- As condicionais **avaliam uma condição** e dependendo do seu resultado é feita **uma opção ou outra**. As condições devem ser específicas e **usar expressões lógicas**.



3. Estruturas Algorítmicas

- Na vida real **tomamos decisões** a partir da análise de algumas condições.



3. Estruturas Algorítmicas

- Na vida real **tomamos decisões** a partir da análise de algumas condições.
 - Exemplo: **Se** eu tiver pelo menos R\$ 50,00 na conta corrente, **então** irei ao cinema hoje à noite.



3. Estruturas Algorítmicas

- Na vida real **tomamos decisões** a partir da análise de algumas condições.
 - Exemplo: **Se** eu tiver pelo menos R\$ 50,00 na conta corrente, **então** irei ao cinema hoje à noite.
 - Trata-se de uma **expressão lógica**, uma vez que a pergunta: *“Tenho R\$ 50,00 sobrando?”* Pode ser respondida com um *“Sim”* ou com um *“Não”*.



3. Estruturas Algorítmicas

- Assim como na vida real, nos algoritmos (e nas linguagens de programação), um determinado **bloco básico de comandos será executado ou não**, dependendo da avaliação de expressões lógico-aritmético-relacionais.
- A isso chamados de **algoritmo com alternativa**.



3. Estruturas Algorítmicas

- As estruturas condicionais podem ser classificadas como:
 - Simples (**SE-ENTÃO**)
 - Duplas (**SE-ENTÃO SENÃO**)

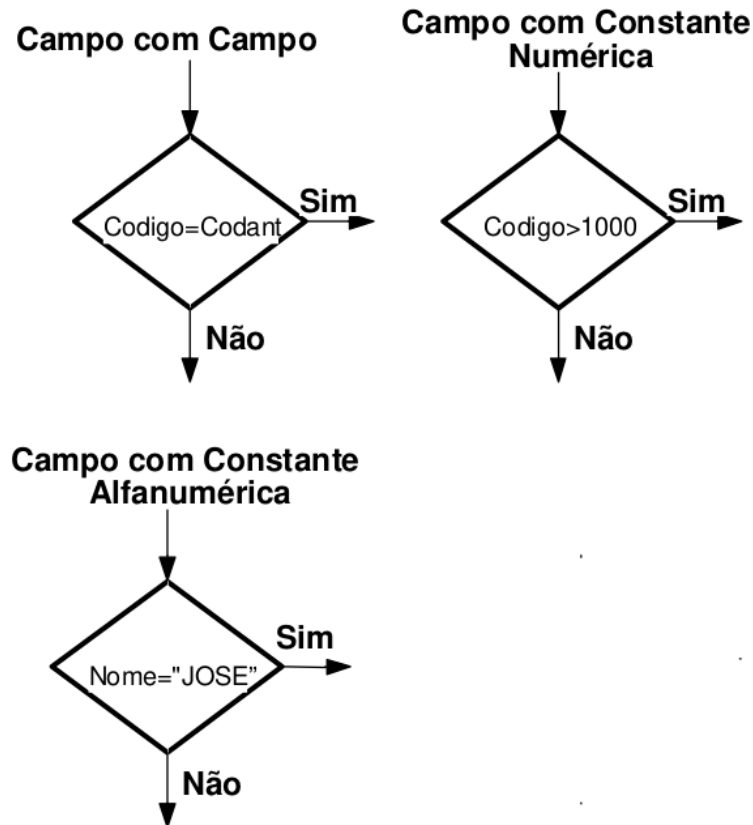


3. Estruturas Algorítmicas

- As estruturas condicionais podem ser classificadas como:
 - Simples (**SE-ENTÃO**)
 - Duplas (**SE-ENTÃO SENÃO**)
- Condicionais são representadas como losângos nos fluxogramas.



3. Estruturas Algorítmicas





3. Estruturas Algorítmicas

Exemplo 1: Escreva um algoritmo para calcular a **área de um círculo** usando como **entrada** o valor do raio, que deve ser positivo. Em seguida o algoritmo deve imprimir o valor da área na tela do computador. Se o raio não for positivo, uma mensagem de erro deve ser exibida e **o raio não é calculado**.



3. Estruturas Algorítmicas

Exemplo 1:

```
Algoritmo CalculaAreaCirculo
  IMPRIMIR "Entre com o raio do círculo:"
  LER raio
  SE raio > 0 ENTÃO
    area <- 3.14 * raio2
    IMPRIMIR "A área do círculo é ", area
  SENÃO
    IMPRIMIR "Raio não pode ser nulo ou negativo!"
  FIM-SE
Fim-Algoritmo
```

3. Estruturas Algorítmicas

Exemplo 1:

Algoritmo CalculaAreaCirculo

IMPRIMIR "Entre com o raio do círculo:"

LER raio

SE raio > 0 **ENTÃO**

area $\leftarrow 3.14 * \text{raio}^2$

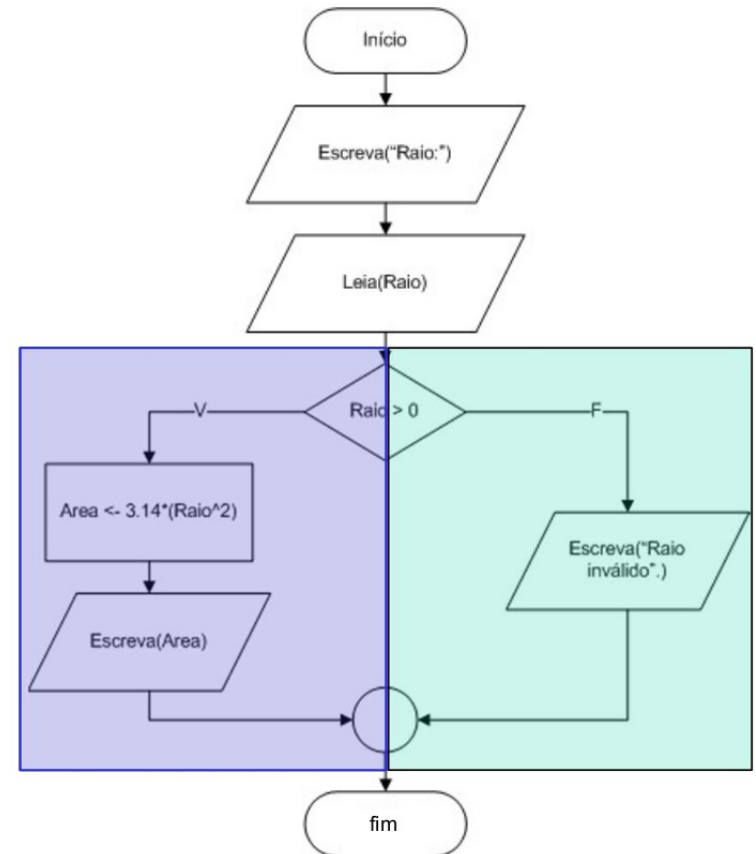
IMPRIMIR "A área do círculo é ", area

SENÃO

IMPRIMIR "Raio não pode ser nulo ou negativo!"

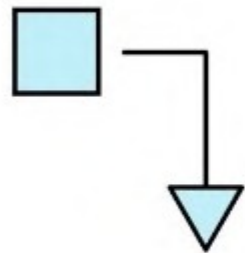
FIM-SE

Fim-Algoritmo



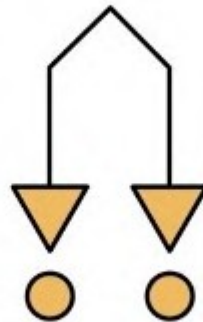
3. Estruturas Algorítmicas

- Nos algoritmos temos 3 estruturas básicas de controle. São elas:



Sequenciais

Condicionais





3. Estruturas Algorítmicas

- Semelhante aos comandos de decisão, os comandos de repetição executam enquanto uma condição for verdadeira.



3. Estruturas Algorítmicas

- Semelhante aos comandos de decisão, os comandos de repetição executam enquanto uma condição for verdadeira.
- Utilizamos os comandos de repetição quando desejamos que um **determinado conjunto de instruções** um número definido ou indefinido de vezes.



3. Estruturas Algorítmicas

- Em algoritmos e linguagens de programação, existem diversos modelos de comandos de repetição:
 - enquanto <condição> faça <comandos>
 - até que <condição> faça <comandos>
 - faça <comandos> enquanto <condição>
 - faça <comandos> até que <condição>



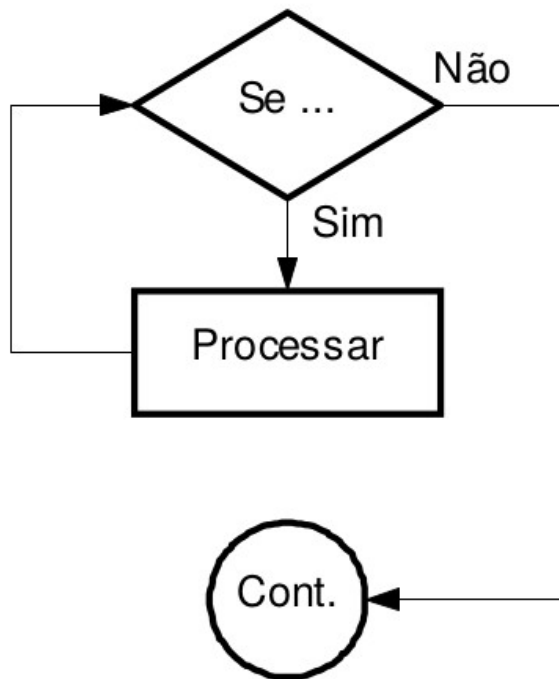
3. Estruturas Algorítmicas

3.1 enquanto <condição> faça <comandos>

- Neste caso, o bloco de operações será executado **enquanto a <condição> for verdadeira**. O teste da condição será **sempre** realizado antes de qualquer operação.
 - Enquanto a condição for verdadeira o processo se repete.
 - Podemos utilizar essa estrutura para trabalharmos com contadores.
-

3. Estruturas Algorítmicas

3.1 enquanto <condição> faça <comandos>



```
Algoritmo ExemploEnquanto  
  ENQUANTO <condição> FAÇA  
    processar  
  FIM-ENQUANTO  
Fim-Algoritmo
```

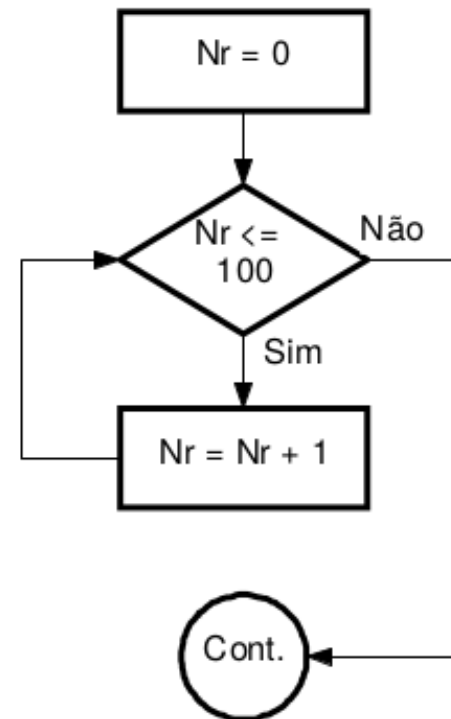


3. Estruturas Algorítmicas

3.1 enquanto <condição> faça <comandos>

Exemplo: contar até 100

```
Algoritmo ExemploContador  
  VAR nr = 0  
  ENQUANTO nr <= 100 FAÇA  
    nr <- nr + 1  
  FIM-ENQUANTO  
Fim-Algoritmo
```





3. Estruturas Algorítmicas

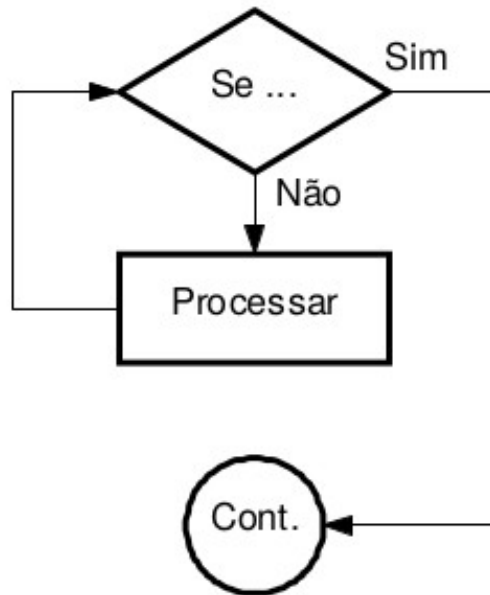
3.1 até que <condição> faça <comandos>

- Neste caso, o bloco de operações será executado **até que a condição seja satisfeita**, ou seja, somente executará os comandos **enquanto a condição for falsa**.



3. Estruturas Algorítmicas

3.2 até que <condição> faça <comandos>



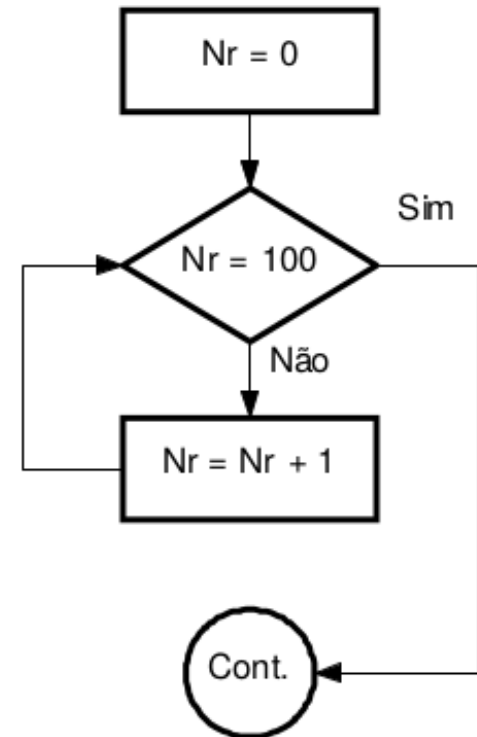
```
Algoritmo ExemploAtéQue
  ATÉ QUE <condição> FAÇA
    processar
  FIM-ATÉ-QUE
Fim-Algoritmo
```


3. Estruturas Algorítmicas

3.2 até que <condição> faça <comandos>

Exemplo: contar até 100

```
Algoritmo ExemploAtéQueContador  
  VAR nr = 0  
  ATÉ QUE nr = 100 FAÇA  
    nr <- nr + 1  
  FIM-ATÉ-QUE  
Fim-Algoritmo
```





3. Estruturas Algorítmicas

3.2 até que <condição> faça <comandos>

Algoritmo ExemploAtéQueContador

VAR nr = 0

ATÉ QUE nr = 100 **FAÇA**

nr ← nr + 1

FIM-ATÉ-QUE

Fim-Algoritmo

Algoritmo ExemploContador

VAR nr = 0

ENQUANTO nr ≤ 100 **FAÇA**

nr ← nr + 1

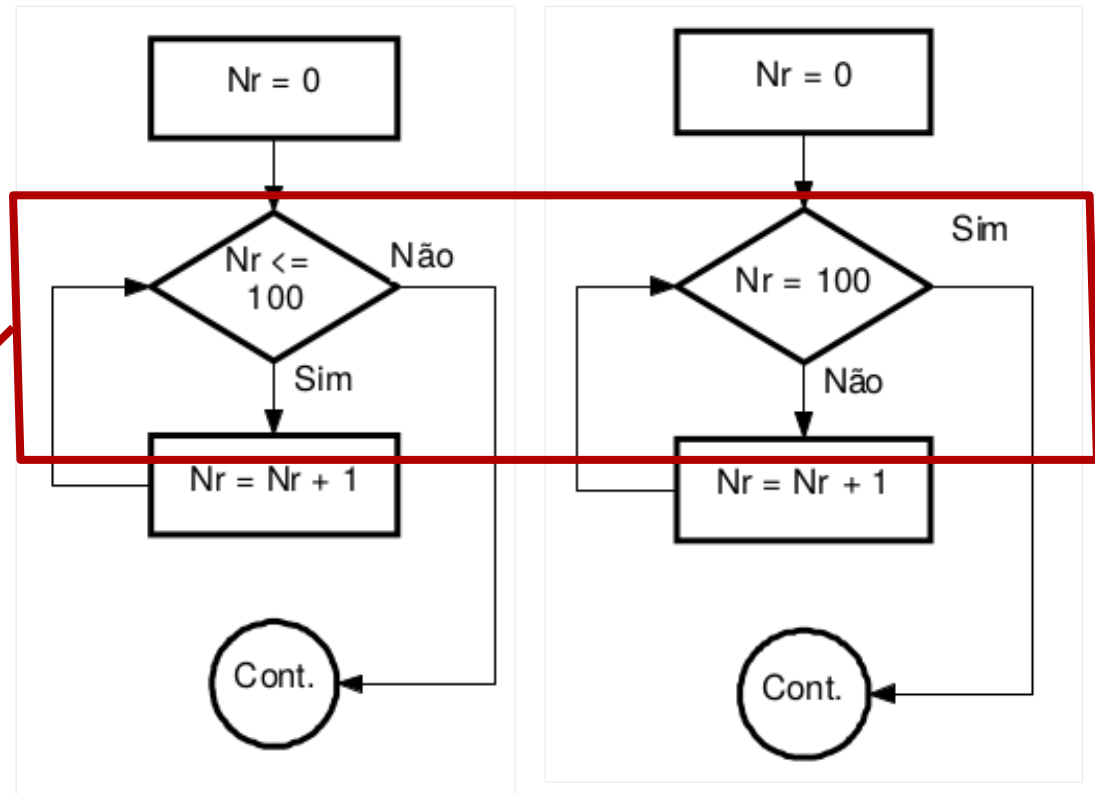
FIM-ENQUANTO

Fim-Algoritmo

3. Estruturas Algorítmicas

3.2 até que <condição> faça <comandos>

Como já comentamos anteriormente, é possível escrever com passos diferentes para resolver o mesmo o problema e chegar no mesmo resultado





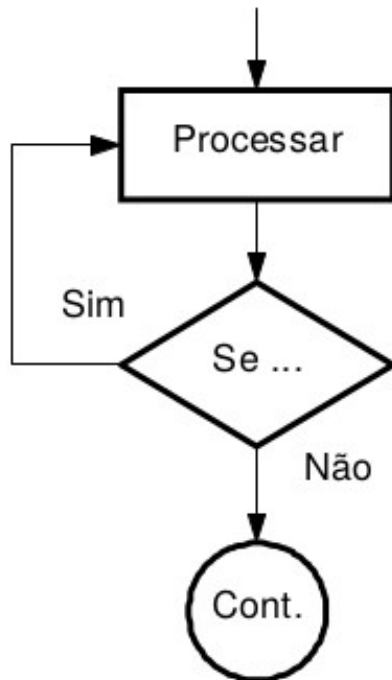
3. Estruturas Algorítmicas

3.3 faça <comandos> enquanto <condição>

- Neste caso primeiro são executados os comandos, e somente depois é realizado o teste da condição.
- Se a condição for verdadeira, os comandos são executados novamente, caso seja falso é encerrado o comando de repetição.
- Ou seja, os comandos são executados ao menos uma vez.

3. Estruturas Algorítmicas

3.3 faça <comandos> enquanto <condição>



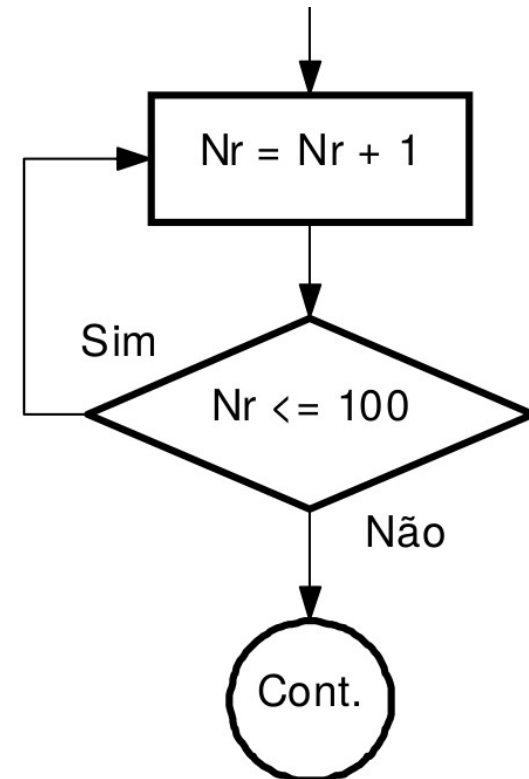
```
Algoritmo FaçaEnquanto  
  FAÇA  
    processar  
  ENQUANTO <condição>  
Fim-Algoritmo
```

3. Estruturas Algorítmicas

3.3 faça <comandos> enquanto <condição>

Exemplo: contar até 100

```
Algoritmo ExemploFacaEnquantoContador  
  VAR nr = 0  
  FAÇA  
    nr <- nr + 1  
  ENQUANTO nr <= 100  
Fim-Algoritmo
```





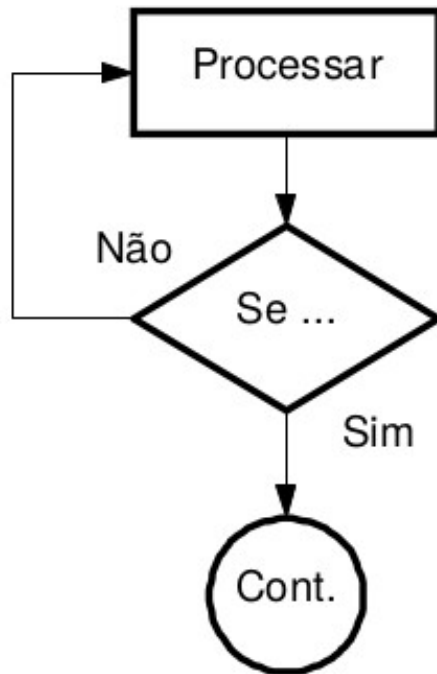
3. Estruturas Algorítmicas

3.4 faça <comandos> até que <condição>

- Novamente, primeiro são executados os comandos, e somente depois é realizado o teste da condição.
- Se a condição for verdadeira, os comandos não são executados novamente, caso seja falso os comandos são executados novamente.
- Ou seja, os comandos são executados ao menos uma vez.

3. Estruturas Algorítmicas

3.3 faça <comandos> até que <condição>



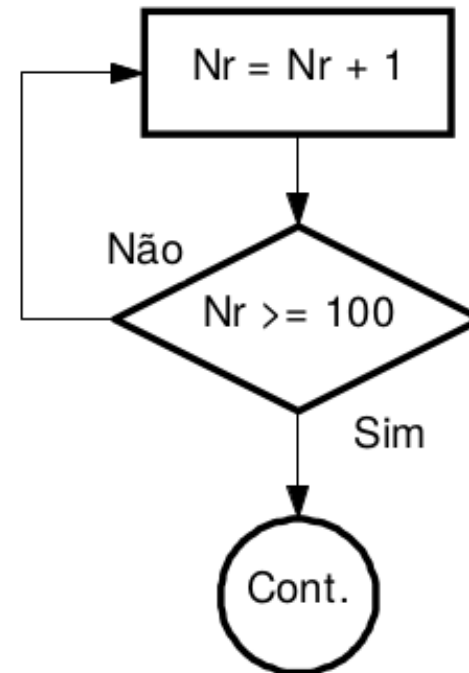
```
Algoritmo FaçaAtéQue  
  FAÇA  
    processar  
  ATÉ-QUE <condição>  
Fim-Algoritmo
```


3. Estruturas Algorítmicas

3.3 faça <comandos> enquanto <condição>

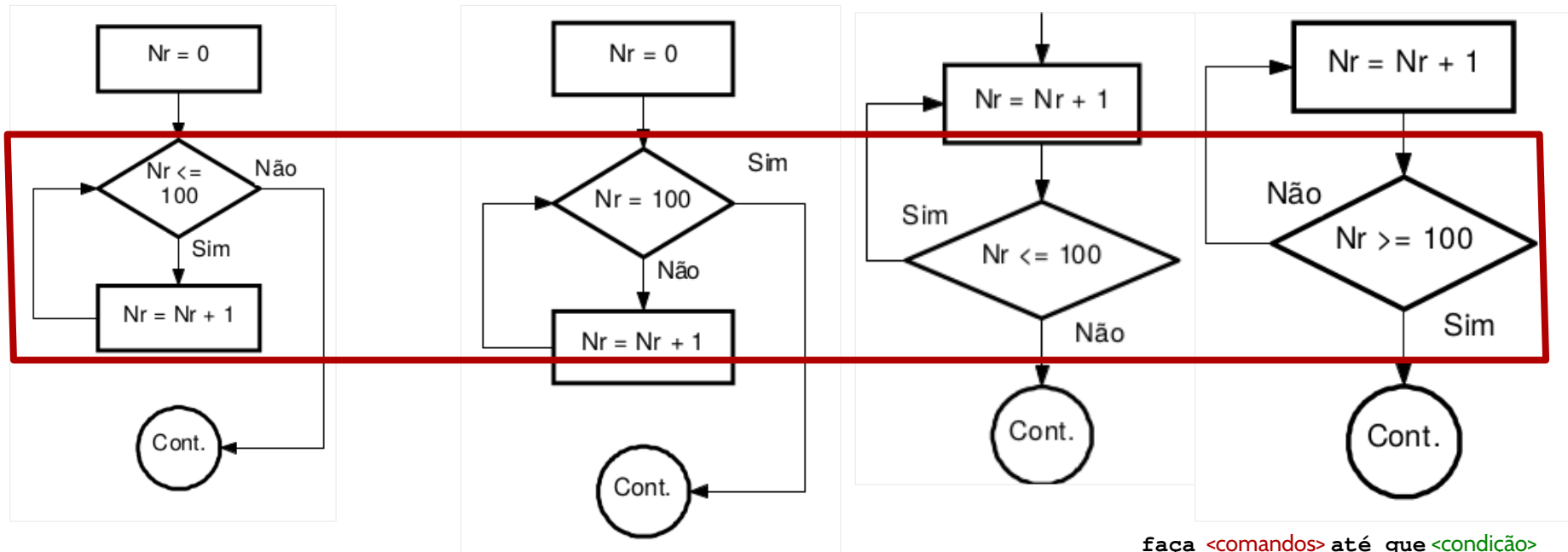
Exemplo: contar até 100

```
Algoritmo ExemploFaçaAtéQueContador  
  VAR nr = 0  
  FAÇA  
    nr <- nr + 1  
  ATÉ QUE nr >= 100  
Fim-Algoritmo
```



3. Estruturas Algorítmicas

Comparação das estruturas de repetição



faça <comandos> até que <condição>

enquanto <condição> faça <comandos>

até que <condição> faça <comandos>

faça <comandos> enquanto <condição>



Exemplo:

E1. Escreva um algoritmo que solicita ao usuário um valor inteiro positivo N e imprime na tela do computador todos os número inteiros de 0 a N .



Exemplo:

E1. Escreva um algoritmo que solicita ao usuário um valor inteiro positivo N e imprime na tela do computador todos os número inteiros de 0 a N.

```
Algoritmo ExemploMostraNNumeros
  VAR n = 0
  VAR c = 0
  IMPRIMIR "Entre com o número N:"
  LER n
  ENQUANTO c <= n FAÇA
    IMPRIMIR "Contador = ", c
    c = c + 1
  FIM-ENQUANTO
Fim-Algoritmo
```



Exemplo:

E2. Escreva um algoritmo que solicita diversos valores reais positivos, calcula e imprime a soma acumulada dos números digitados. O programa continua solicitando os números até que o valor “-1” seja digitado como entrada pelo usuário.



Exemplo:

E2. Escreva um algoritmo que solicita diversos valores reais positivos, calcula e imprime a soma acumulada dos números digitados. O programa continua solicitando os números até que o valor “-1” seja digitado como entrada pelo usuário.

```
Algoritmo ExemploMostraNNumeros
  VAR input_num = 0
  VAR sum = 0
  IMPRIMIR "Entre com um número"
  LER input_num
  ENQUANTO input_num != -1 FAÇA
    sum = sum + input_num
    IMPRIMIR "Somatório parcial é", sum
    IMPRIMIR "Entre com outro número"
    LER input_num
  FIM-ENQUANTO
Fim-Algoritmo
```



Exemplo:

E2. Escreva um algoritmo que solicita diversos valores reais positivos, calcula e imprime a soma acumulada dos números digitados. O programa continua solicitando os números até que o valor “-1” seja digitado como entrada pelo usuário.

```
Algoritmo ExemploMostraNNumeros
  VAR input_num = 0
  VAR sum = 0
  IMPRIMIR "Entre com um número"
  LER input_num
  ENQUANTO input_num != -1 FAÇA
    sum = sum + input_num
    IMPRIMIR "Somatório parcial é", sum
    IMPRIMIR "Entre com outro número"
    LER input_num
  FIM-ENQUANTO
Fim-Algoritmo
```

≠



Exercício

E1. Escreva um algoritmo que recebe do usuário um valor inteiro positivo “n” e imprime na tela todos números inteiros positivos de “n” até 0.



Exercício

E1. Escreva um algoritmo que recebe do usuário um valor inteiro positivo “n” e imprime na tela todos números inteiros positivos de “n” até 0.

```
Algoritmo ExemploMostraNNumeros
  VAR n = 0
  VAR i = 0
  IMPRIMIR "Entre o valor de N:"
  LER n
  i = n
  ENQUANTO i >= 0 FAÇA
    IMPRIMIR "i=", i
    I = i - 1
  FIM-ENQUANTO
Fim-Algoritmo
```



A word cloud centered around the word "THANK YOU". Other prominent words include "GRACIAS", "ARIGATO", "SHUKURIA", "JUSPAXAR", "DANKSCHEEN", "TASHAKKURATU", "YAQHANYELAY", "SUKSAMA", "EKHMET", "TINGKI", "BİYAN", "SHUKRIA", "GOZAIMASHITA", "BECHARISTO", "KOMAPSUMNDA", "MAAKE", "GRAZIE", "MEHRBANI", "PALMES", "BOLZİN", and "MERCİ". The words are arranged in various sizes and orientations, creating a dense, colorful collage.



Dúvidas?

