

**Aula**



# **Algoritmo e Estruturas de Dados**



## *Expressões Lógicas*

É comum nos algoritmos surgirem situações em que a execução de uma ação, ou sequência de subações, está sujeita a uma certa condição. Esta condição é representada no texto do algoritmo por meio de uma expressão lógica.

Denomina-se expressão lógica a expressão cujos operadores são lógicos e cujos operandos são relações, constantes e/ou variáveis do tipo lógico.

## *Operadores Relacionais*

Uma expressão relacional, ou simplesmente relação, é uma comparação realizada entre dois, valores de mesmo tipo básico. Estes valores são representados na relação através de constantes, variáveis ou expressões aritméticas, estas últimas para o caso de valores numéricos.

Os operadores relacionais, que indicam a comparação a ser realizada entre os termos da relação, são conhecidos da Matemática, a saber:

**= igual a**

**≠ diferente de**

**> maior que**

**< menor que**

**≥ maior ou igual a**

**≤ menor ou igual a**

O resultado obtido de uma relação é sempre um valor lógico.

# PORTUGOL – Comandos Básicos

## Operadores Relacionais

Eis algumas relações:

a)  $A \neq B$ ;

b)  $NOME = "JOÃO"$ ;

c)  $B^2 - 4 \times A \times C < 0$ ;

d)  $X = 1$ .

VARIÁVEIS					RELAÇÕES		
X	Y	Z	COR	NOME	$X^2 + Y > Z$	$COR = "AZUL"$	$NOME \neq "JOSÉ"$
1	2	5	"AZUL"	"PAULO"	falso	verdadeiro	verdadeiro
4	3	1	"VERDE"	"JOSÉ"	verdadeiro	falso	falso
1	1	2	"BRANCO"	"PEDRO"	falso	falso	verdadeiro
1	2	1	"AZUL"	"JOSÉ"	verdadeiro	verdadeiro	falso

# PORTUGOL – Comandos Básicos

## *Operadores Relacionais*

Eis algumas relações:

VARIÁVEIS				RELAÇÕES		
A	B	NOME	PROFISSÃO	$A + 1 \geq \sqrt{B}$	NOME $\neq$ "ANA"	PROFISSÃO $=$ "MÉDICO"
3	16	"MIRIAM"	"ADVOGADO"	Verdadeiro	Verdadeiro	Falso
5	64	"PEDRO"	"MÉDICO"	Falso	Verdadeiro	Verdadeiro
2.5	9	"ANA"	"PROFESSOR"	Verdadeiro	Falso	Falso

# Operadores Lógicos

---

- Viabiliza a utilização de mais de uma condição na mesma instrução **se**;
- **.e.** , **.ou.** e **.não.**
- Pode evitar o uso de instruções **se** encadeadas



# Operador Lógico .e.

- Utilizado quando dois ou mais relacionamentos lógicos de uma determinada condição necessitam ser verdadeiros.

<b>Condição1</b>	<b>Condição2</b>	<b>Resultado</b>
Falsa	Falsa	<i>Falso</i>
Verdadeira	Falsa	<i>Falso</i>
Falsa	Verdadeira	<i>Falso</i>
Verdadeira	Verdadeira	<i>Verdadeiro</i>



# Operador Lógico .e.

---

```
se <condicao1> .e. <condicao2> então  
    instrucoes executadas se condicao1 e  
    condicao2 verdadeiras  
fim_se
```



# Operador Lógico .e.

## Exemplo

```
programa TESTA_LOGICA_E
```

```
var
```

```
    NUMERO: inteiro
```

```
início
```

```
    leia NUMERO
```

```
    se (NUMERO >= 20) .e. (NUMERO <= 90) então
```

```
        escreva "O número está na faixa de 20 a  
        90"
```

```
    senão
```

```
        escreva "O número está fora da faixa de  
        20 a 90"
```

```
    fim_se
```

```
fim
```



# Operador Lógico .ou.

- Utilizado quando pelo menos um dos relacionamentos lógicos de uma determinada condição necessita ser verdadeiro.

<b>Condição1</b>	<b>Condição2</b>	<b>Resultado</b>
Falsa	Falsa	<i>Falso</i>
Verdadeira	Falsa	<i>Verdadeiro</i>
Falsa	Verdadeira	<i>Verdadeiro</i>
Verdadeira	Verdadeira	<i>Verdadeiro</i>

# Operador Lógico .ou.

---


```
se <condicao1> .ou. <condicao2> então  
    instrucoes executadas se condicao1  
    verdadeira ou condicao2 verdadeira  
fim_se
```

# Operador Lógico .ou.

## Exemplo

---

```
programa TESTA_LOGICA_OU
var
    SEXO: caractere
início
    leia SEXO
    se (SEXO = "masculino") .ou. (SEXO =
        "feminino") então
        escreva "O seu sexo é valido"
    senão
        escreva "O seu sexo é inválido"
    fim_se
fim
```



# Operador Lógico .não.

- Utilizado quando houver necessidade de se **inverter** o resultado lógico de uma condição.
- Se a condição for verdadeira será considerada falsa e se for falsa será considerada verdadeira.

Condição	Resultado
Verdadeira	<i>Falso</i>
Falsa	<i>Verdadeiro</i>



# Operador Lógico .não.

---

**se .não.<condicao1> então**

instrucoes executadas se condicao for  
falsa

**fim\_se**



# Operador Lógico . não.

## Exemplo

```
programa TESTA_LOGICA_NAO
var
    A, B, C, D: inteiro
início
    leia A, B, D
    se .não. (D > 5) então
        C <- (A + B) * D
    senão
        C <- (A - B) * D
    fim_se
    escreva C
fim
```

# Tabela Verdade

p	q	not p	not q	p or q	p e q	not p e not q	not p or not q
V	V	F	F	V	V	F	F
V	F	F	V	V	F	F	V
F	V	V	F	V	F	F	V
F	F	V	V	F	F	V	V

**Legenda:**

not = Não

or = Ou

e = E



## *Expressões lógicas*

Eis algumas expressões lógicas:

- a)  $A + B = 0$  e  $C \neq 1$ ;
- b) `TESTE` ou  $A \times C > B$ ;
- c) não `TESTE` e `COR = "AZUL"`.

Onde A, B e C são variáveis numéricas, `TESTE` é uma variável lógica e `COR` uma variável literal.


O resultado obtido da avaliação de uma expressão lógica é sempre um valor lógico, isto é, um falso ou verdadeiro.

Por este motivo, pode-se considerar uma única relação como sendo uma expressão lógica.

## *Prioridades*

Em alguns casos, conforme os valores envolvidos, a ordem em que são efetuadas as operações lógicas afeta o resultado final. Assim como acontece entre as operações aritméticas, também existe uma relação de prioridade entre os operadores lógicos.

Também nas expressões lógicas vários níveis de parênteses podem ser utilizados com a finalidade de estabelecer uma nova ordem de execução entre os operadores lógicos.



# PORTUGOL – Comandos Básicos

## *Prioridades – Exemplos*

Prioridade	Operador
1	Parênteses e Funções
2	Aritmético (* /)
3	Aritmético (+ -)
4	Relacional
5	Não
6	E
7	Ou

a)  $A = 1 \text{ e } (B + C \# 0 \text{ ou } K \leq 2);$

b)  $\text{não } (TOTAL \geq 2 \text{ e } A \# B) \text{ ou } \text{TESTE}.$

## *Expressões Literais*

Uma expressão literal é aquela formada por operadores literais e operandos que são constantes e/ou variáveis do tipo literal.

Supondo que A e B são variáveis literais e que o símbolo “|” é um operador de concatenação de literais, a expressão A|B fornece como resultado um único literal formado pelo conteúdo de A seguido do conteúdo de B.


Exemplo: Se A contém o literal “BOLA” e B contém o literal “PRETA”, o valor fornecido pela expressão A|B é o literal “BOLAPRETA”.

## *Comandos de Entrada e Saída*

Sabe-se que as unidades de entrada e saída são dispositivos que possibilitam a comunicação entre o usuário e o computador.

Por exemplo, através de um teclado. o usuário consegue dar entrada ao programa e aos dados na memória do computador.

Por sua vez, o computador pode emitir os resultados e outras mensagens para o usuário através de uma impressora de linhas.



## *Comandos de Entrada e Saída*

Seja a seguinte situação: início da execução de um programa que se encontra armazenado na memória principal do computador. Como e quem determina o momento da entrada dos dados para o programa e a saída dos resultados obtidos para o usuário?

Isto é tarefa do programador e ele assim o faz quando, no desenvolvimento do algoritmo, descreve as ações a serem executadas pelo computador.

Os comandos de entrada e saída são as ferramentas para esta finalidade.





# PORTUGOL – Comandos Básicos

---


## *Comandos de Entrada e Saída*

A sintaxe do comando de entrada é:

Leia lista-de-identificadores

Onde:

Leia	é uma palavra-chave;
Lista-de-identificadores	são os nomes das variáveis, separados por vírgula, nas quais serão armazenados os valores provenientes do meio de entrada.



## *Comandos de Entrada e Saída - Exemplo*

Supondo que NOTA e NUM são variáveis do tipo numérico, o comando: ***leia NOTA, NUM*** indica que dois valores numéricos serão lidos de uma unidade de entrada, quando este comando for executado.

Os valores serão armazenados nas posições de memória (variáveis) identificadas pelos nomes NOTA e NUM.





## *Comandos de Entrada e Saída - Exemplo*

Analogamente, um comando de saída tem a forma geral:

**escreva** lista-de-identificadores e / ou constantes


Onde:

escreva

é uma palavra-chave;

Lista-de-identificadores

são os nomes das variáveis, cujos conteúdos serão mostrados ao usuário através de um meio de saída. Além dos conteúdos das variáveis, o valor de uma constante pode ser emitido diretamente.



# PORTUGOL – Comandos Básicos

---

## *Comandos de Entrada e Saída - Exemplo*

- a) leia X;
- b) leia NOME,N,Y;
- c) Escreva K,SOMA;
- d) Escreva 21, "NOME",N;
- c) Escreva "TABELA DE PREÇOS".



# Exercício

Faça um algoritmo em português estruturado, onde o mesmo receba dois valores e depois exiba a soma desses valores.

# Exercício

Faça um algoritmo em português estruturado, onde o mesmo receba dois valores e depois exiba a soma desses valores.

```
programa soma
var
    x : inteiro
    a : inteiro
    b : inteiro
inicio
    leia a
    leia b
    x <- a + b
    escreva x
fim
```

# Estruturas de Controle

## Tomada de Decisão

- Desvio Condicional Simples (**se... então... fim\_se.**)
  - Sendo a condição **verdadeira** serão executadas as instruções entre a instrução **então** e a instrução **fim\_se**.
  - Sendo a condição **falsa** serão executadas as instruções após a instrução **fim\_se**.

**se** <condicao> **então**

instrucoes para condicao verdadeira

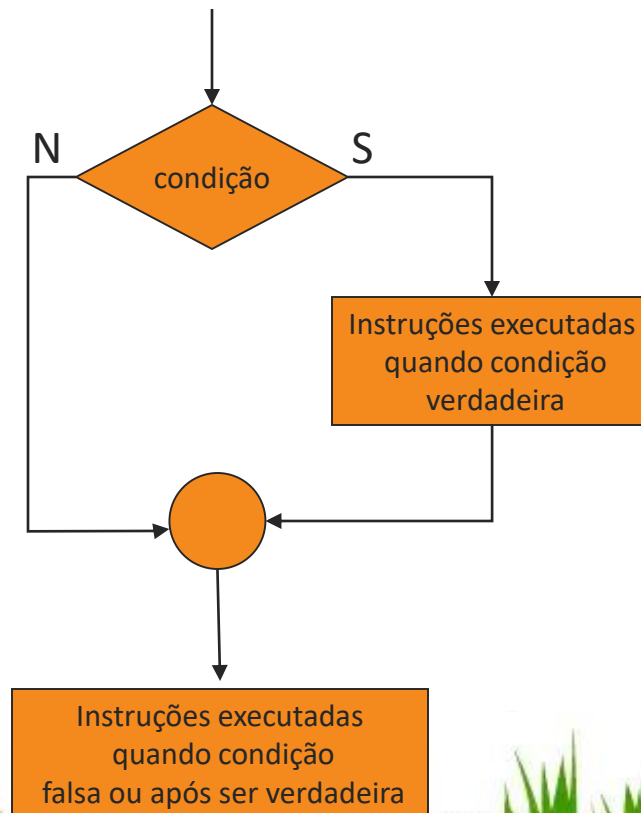
**fim\_se**

instrucoes para condicao falsa ou após  
ser verdadeira

# Estruturas de Controle

## Tomada de Decisão

- Desvio Condicional Simples



# Desvio condicional Simples

## Exemplo

- Leia dois valores numéricos, somá-los e apresentar o resultado caso este seja maior que 10.

```
programa SOMANUMEROS
var
    X: inteiro
    A: inteiro
    B: inteiro
início
    leia A
    leia B
     $X \leftarrow A + B$ 
    se (X > 10) entao
        escreva X
    fimse
fim
```

# Estruturas de Controle

## Tomada de Decisão

- Desvio Condicional Composto (**se... então... senão... fim\_se.**)
  - Sendo a condição **verdadeira** serão executadas as instruções entre a instrução **então** e a instrução **senão** .
  - Sendo a condição **falsa** serão executadas as instruções entre a instrução **senão** e a instrução **fim\_se** .

**se** <condicao> **entao**

instrucoes para condicao verdadeira

**senão**

instrucoes para condicao falsa

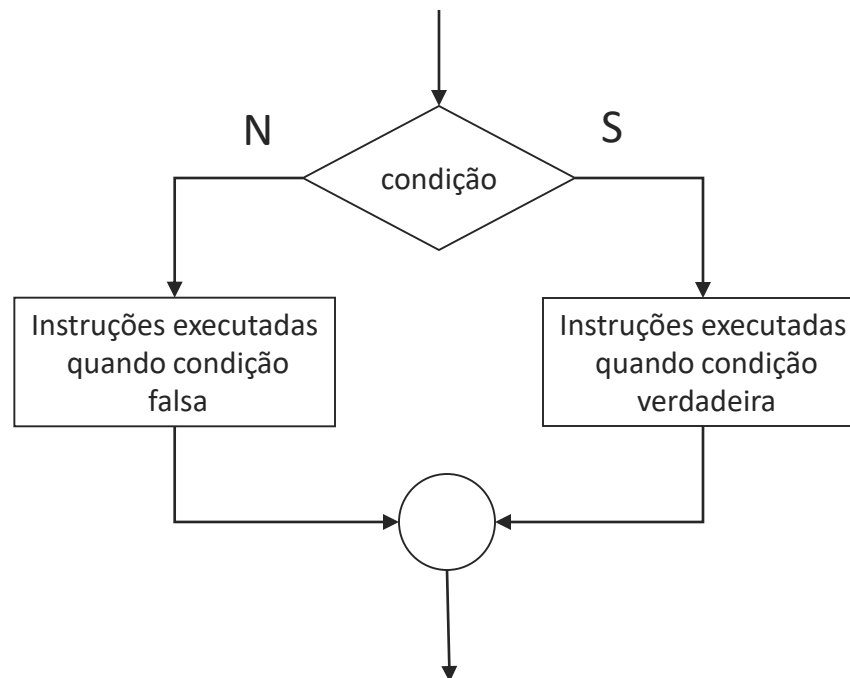
**fimse**



# Estruturas de Controle

## Tomada de Decisão

- Desvio Condicional Composto



# Desvio condicional Composto

## Exemplo

- Leia dois valores numéricos e efetuar a adição. Caso o valor somado seja maior ou igual a 10, deverá ser apresentado somando a ele mais 5; caso o valor somado não seja maior ou igual a 10, ele deverá ser apresentado subtraindo 7.

```
programa SOMANUMEROS
var
    X: inteiro
    A: inteiro
    B: inteiro
    R: inteiro
início
    leia A,B
    X ← A + B
    se (X >= 10) então
        R ← X + 5
    senao
        R ← X - 7
    fimse
    escreva R
fim
```

# Estruturas de Controle

## Tomada de Decisão

---

- Desvio Condicional Encadeado ou Aninhado
  - Equivale a uma estrutura condicional dependente de outra, ou ainda, dentro da outra.

**se** *<condicao1>* **entao**

instrucoes para condicao1 verdadeira

**senao**

**se** *<condicao2>* **entao**

instrucoes para condicao2

verdadeira, porém condicao1 falsa

**senao**

instrucoes para condicao1 e condicao2 falsa

**fimse**

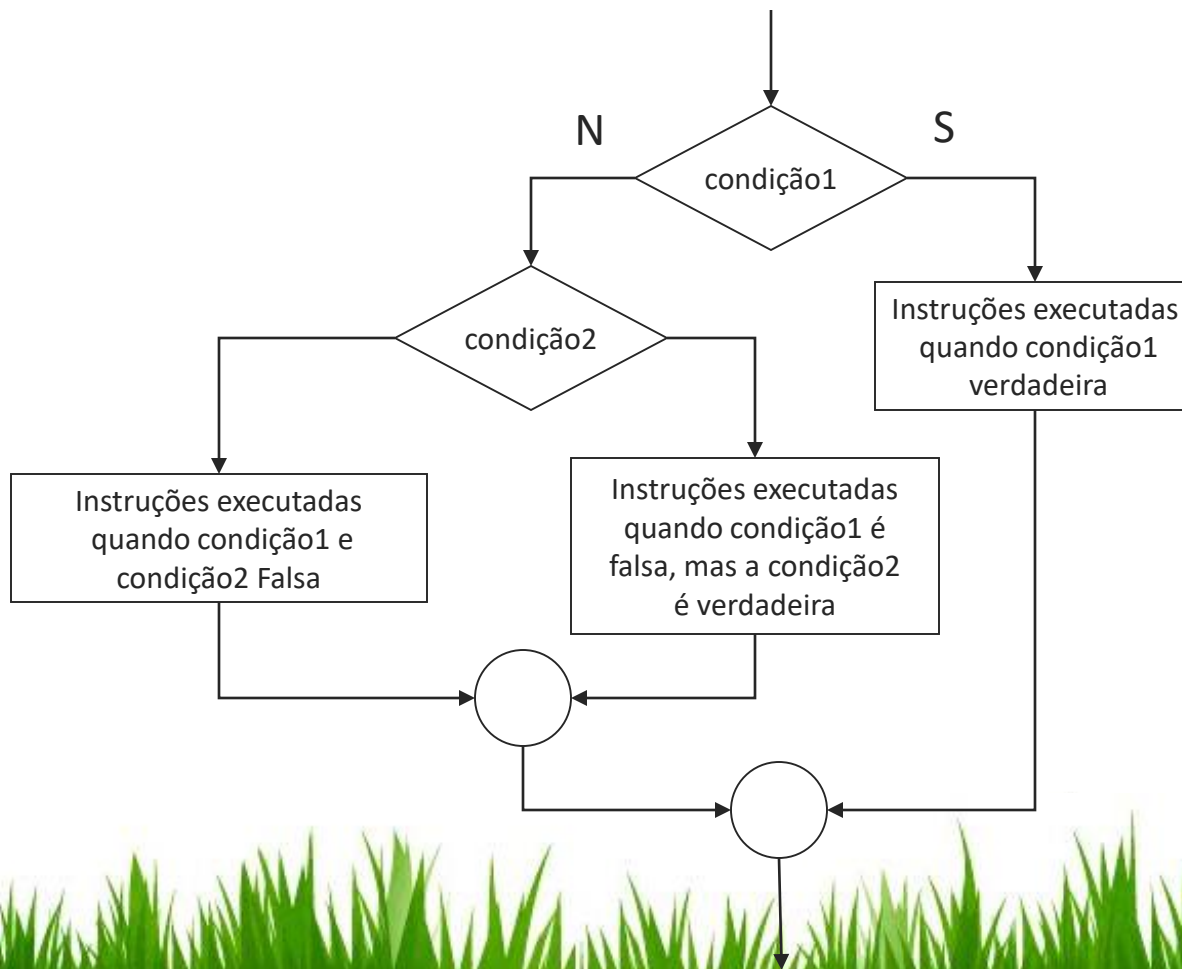
**fimse**



# Estruturas de Controle

## Tomada de Decisão

- Desvio Condicional Encadeado ou Aninhado



# Desvio condicional Encadeado Exemplo

- Elaborar um programa que efetue o cálculo do reajuste de salário de um funcionário. Considere que o funcionário deverá receber um reajuste de 15% caso seu salário seja menor que 500. Se o salário for maior ou igual a 500, mas menor ou igual a 1000, seu reajuste será de 10%; caso seja ainda maior que 1000 o reajuste deverá ser de 5%.

```
programa REAJUSTE_SALARIO
var
    NOVOSALARIO: real
    SALARIO: real
início
    leia (SALARIO)
    se (SALARIO < 500) entao
        NOVOSALARIO ← SALARIO * 1.15
    senao
        se (SALARIO <= 1000) entao
            NOVOSALARIO ← SALARIO * 1.10
        senao
            NOVOSALARIO ← SALARIO * 1.05
        fimse
    fimse
    escreva (NOVOSALARIO)
fim
```

# Exercícios

2. Indique o valor da variável X nos trechos de programas abaixo. Considere A=2, B=3, C=5 e D=9.

a) **se .não. (D > 5) então**  
     $X \leftarrow (A + B) * D$   
**senão**  
     $X \leftarrow (A * B) / 3$   
**fim\_se**  
**escreva X**      **2**

b) **se (A > 2) .e. (B < 7) então**  
     $X \leftarrow (A + 2) * (B - 2)$   
**senão**  
     $X \leftarrow (7 + B) / 5 * (C + D)$   
**fim\_se**  
**escreva X**      **28**

c) **se (A = 2) .ou. (B < 7)**  
**então**  
     $X \leftarrow (A + 2) * (B - 2)$   
**senão**  
     $X \leftarrow (A + B) / D * (C + D)$   
**fim\_se**  
**escreva X**      **4**

d) **se (A > 2) .ou. .não. (B < 7) então**  
     $X \leftarrow A + B - 2$   
**senão**  
     $X \leftarrow A - B$   
**fim\_se**  
**escreva X**      **-1**

e) **se .não. (A > 2) .ou. .não. (B < 7) então**  
     $X \leftarrow A + B$   
**senão**  
     $X \leftarrow A / B$   
**fim\_se**  
**escreva X**      **5**

f) **se .não. (A > 3) .e. .não. (B < 5) então**  
     $X \leftarrow A + D$   
**senão**  
     $X \leftarrow D / B$   
**fim\_se**  
**escreva X**      **3**

# Exercícios

g) **se** (C >= 2) **.e.** (B <= 7) **então**  
    X ← (A + D) / 2  
**senão**  
    X ← D \* C  
**fim\_se**  
**escreva** X           **5.5**

h) **se** (A >= 2) **.ou.** (C <= 1) **então**  
    X ← (A \* D) / 2  
**senão**  
    X ← D \* C  
**fim\_se**  
**escreva** X           **9**