



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE

# COMANDOS DE DESVIO CONDICIONAL (SIMPLES E ENCADEADOS) E OPERADORES LÓGICOS

Disciplina: Algoritmos

Professora: Alba Lopes

[alba.lopes@ifrn.edu.br](mailto:alba.lopes@ifrn.edu.br)

<http://docente.ifrn.edu.br/albalopes>

*Baseado nos slides de aula do prof. Jalerson Lima*

# INTRODUÇÃO

- O comando condicional “se” executa um bloco de código caso uma expressão lógica seja verdadeira
- Sintaxe:

```
se <expressão-lógica> entao  
    <sequencia-de-comandos>  
fimse
```



# EXPRESSÕES LÓGICAS

- As expressões lógicas são avaliações que retornam VERDADEIRO ou FALSO
  - Ou seja, retornam um valor lógico
- É comum expressões lógicas usarem operadores relacionais



# OPERADORES RELACIONAIS

Operador	Nome	Descrição
=	Igual	Verifica se um valor é igual a outro
<	Menor	Verifica se um valor é menor que outro
>	Maior	Verifica se um valor é maior que outro
<=	Menor ou igual	Verifica se um valor é menor ou igual a outro
>=	Maior ou igual	Verifica se um valor é maior ou igual a outro
<>	Diferente	Verifica se um valor é diferente de outro



# EXEMPLOS COM OPERADORES RELACIONAIS

```
algoritmo "operadores-relacionais"  
var  
inicio  
  escreva(3 = 4)  
  escreva(3 < 4)  
  escreva(3 > 4)  
  escreva(3 <= 4)  
  escreva(3 >= 4)  
  escreva(3 <> 4)  
fimalgoritmo
```



# EXEMPLO

```
algoritmo "situacao-aluno"  
var  
nota : inteiro  
inicio  
escreva("Digite a nota do aluno (0 a 100): ")  
leia(nota)  
se nota >= 60 entao  
    escreva("Aluno aprovado!")  
fimse  
se nota < 60 entao  
    escreva("Aluno reprovado!")  
fimse  
fimalgoritmo
```



## COMANDO SENAO

- Podemos usar o comando “senao” para executar um bloco de código quando a expressão lógica for falsa
- Cada “se” só pode ter um único “senao”, e vice-versa



# MELHORANDO O EXEMPLO

```
algoritmo "situacao-aluno"  
var  
nota : inteiro  
inicio  
escreva("Digite a nota do aluno (0 a 100): ")  
leia(nota)  
se nota >= 60 entao  
    escreva("Aluno aprovado!")  
senao  
    escreva("Aluno reprovado!")  
fimse  
fimalgoritmo
```



# Exercícios



# EXERCÍCIOS

1. Ler dois valores e imprimir o maior dos dois.
2. Ler dois valores e imprimir a diferença do maior para o menor.
3. Desenvolva um algoritmo que leia 3 notas (de 0 a 100), calcule a média aritmética e informe se o aluno passou. A média de aprovação é 70.
4. Desenvolva um algoritmo que, dado um número inteiro, diga se este número é par ou ímpar.



# EXERCÍCIOS

5. Ler 2 números inteiros. Se o segundo for diferente de zero, calcular e imprimir o quociente do primeiro pelo segundo. Caso contrário, imprimir a mensagem: "DIVISÃO POR ZERO".
6. Desenvolva um algoritmo que leia dois números inteiros e deixe-os em ordem crescente.



# COMANDOS CONDICIONAIS ENCADEADOS (OU ANINHADOS)



# RELEMBRANDO COMANDO CONDICIONAL

- O comando de desvio condicional permite executar um bloco de código quando uma expressão lógica for verdadeira

```
se <expressão-lógica> entao
    <bloco de código>
fimse
```

- Um **se** pode estar associado a um **senao**, que executa um bloco de código quando a expressão lógica é falsa

```
se <expressão-lógica> entao
    <bloco-de-código>
senao
    <bloco-de-código>
fimse
```

# INTRODUÇÃO

- Os comandos de desvio condicional podem estar aninhados
  - Ou seja, um comando dentro do outro

```
se <expressão-lógica> entao
    se <expressão-lógica> entao
        <bloco-de-código>
    fimse
fimse
```



# EXEMPLO 1

- Crie um algoritmo que leia a nota de um aluno e mostre se ele está aprovado ( $\geq 60$ ), em recuperação ( $< 60$ ) ou reprovado ( $< 20$ )

```
algoritmo "situacao-aluno"  
var  
nota : inteiro  
inicio  
escreva("Digite uma nota (de 0 a  
100): ")  
leia(nota)
```

```
se nota  $\geq$  60 entao  
    escreva("Aprovado!")  
senao  
    se nota  $\geq$  20 entao  
        escreva("Recuperação!")  
    senao  
        escreva("Reprovado!")  
    fimse  
fimse  
fimalgoritmo
```

## EXEMPLO 2

- O imposto de renda de uma pessoa varia segundo uma tabela.
  - Se o salário for menor do que R\$ 1.000,00, não há imposto
  - Se for entre R\$ 1.000,00 e R\$ 2.200,00, o imposto é de 13% do valor do salário
  - Se for maior do que R\$ 2.200,00, o imposto é de 22%
  - Faça um algoritmo que dado um valor, em reais, correspondente a um salário, informe o valor que será recebido (total menos imposto).





## EXEMPLO 2

```
algoritmo "imposto"  
var  
salario_bruto : real  
salario_liquido : real  
imposto : real  
inicio  
escreva("Digite seu salário: ")  
leia(salario_bruto)  
se salario_bruto < 1000 entao  
    escreval("Não há imposto")
```

```
senao  
    se salario_bruto <= 2200  
entao  
    imposto := salario_bruto *  
0.13  
    senao  
        imposto := salario_bruto *  
0.22  
    fimse  
fimse  
salario_liquido := salario_bruto  
- imposto  
escreval("Seu salário líquido é:  
R$ ", salario_liquido)  
fimalgoritmo
```

## EXEMPLO 2

- Faça um algoritmo que leia o IMC (Índice de Massa Corporal) de uma pessoa.
  - Se o resultado for abaixo de 18,5, informe uma mensagem de que a pessoa está abaixo do peso;
  - Se o resultado for entre 18,5 e 24,99, informe uma mensagem de que a pessoa está com peso normal
  - Se o resultado for acima de 25, informe uma mensagem de que a pessoa está acima do peso.



# OPERADORES LÓGICOS

- Os operadores lógicos são usados nas expressões lógicas
  - Operadores lógicos binários: que combinam duas expressões lógicas – ‘e’, ‘ou’, ‘xou’
  - Operador lógico unário: que precisa de apenas uma expressão lógica – ‘nao’



# OPERADORES LÓGICOS

- O operador lógico 'e' combina duas expressões lógicas, formando uma expressão lógica maior

<expressão-lógica-a> e <expressão-lógica-b>

- A expressão lógica maior vai ser verdadeira quando as duas expressões lógicas forem verdadeiras



# OPERADORES LÓGICOS

## ○ Exemplos:

- Resultam verdadeiro:

$$3 < 4 \text{ e } 6 > 2$$

$$5 > 2 \text{ e } 7 > 1 \text{ e } 4 \geq 4$$

- Resultam falso:

$$10 < 5 \text{ e } 7 > 4$$

$$20 > 10 \text{ e } 4 < 7 \text{ e } 6 < 6$$

# TABELA VERDADE DO E

A	B	A e B
V	V	V
V	F	F
F	V	F
F	F	F



# EXEMPLO 1

- Crie um algoritmo que leia uma nota de 0 a 100 e mostre o conceito relativo à nota:
  - Se a nota estiver entre 80 e 100: conceito A
  - Se a nota estiver entre 60 e 79: conceito B
  - Se a nota estiver entre 40 e 59: conceito C
  - Se a nota estiver entre 20 e 39: conceito D
  - Se a nota for menor que 20: conceito E



# EXEMPLO 1

```
algoritmo "conceito"  
var  
nota : inteiro  
inicio  
escreva("Digite uma nota (de 0 a  
100): ")  
leia(nota)  
se (nota >= 80) e (nota <= 100)  
entao  
    escreva("Conceito A")  
senao  
    se (nota >= 60) e (nota < 80) entao  
        escreva("Conceito B")  
    senao
```

```
        se (nota >= 40) e (nota < 60) entao  
            escreva("Conceito C")  
        senao  
            se (nota >= 20) e (nota < 40)  
entao  
            escreva("Conceito D")  
        senao  
            escreva("Conceito E")  
        fimse  
    fimse  
fimse  
fimse  
fimse  
fimalgoritmo
```



## EXEMPLO 2

- Crie um algoritmo que leia 3 (A, B, C) valores, que representam lados de um triângulo. Um triângulo é válido quando a soma de dois lados é maior que o terceiro lado:
  - $A + B > C$  e  $A + C > B$  e  $B + C > A$



## EXEMPLO 2

algoritmo "lados-triangulo"

var

a, b, c : inteiro

inicio

leia(a)

leia(b)

leia(c)

se  $((a+b) > c)$  e  $((a+c) > b)$  e  $((b+c) > a)$  entao

    escreva("São lados válidos para um triângulo!")

senao

    escreva("São lados inválidos para um triângulo!")

fimse

fimalgoritmo

# OPERADOR LÓGICO OU

- O operador lógico 'ou' também combina duas expressões lógicas, formando uma expressão lógica maior

`<expressão-lógica-a> ou <expressão-lógica-b>`

- A expressão lógica maior vai ser verdadeira quando pelo menos uma das expressões lógicas forem verdadeiras
- Quando as duas expressões forem verdadeiras, o resultado é verdadeiro



# TABELA VERDADE DO OU

A	B	A ou B
F	F	F
F	V	V
V	F	V
V	V	V



# EXEMPLO 1

- Informe se um dado ano (informado pelo usuário) é ou não bissexto. Dica: um ano é bissexto:
  - Se ele for divisível por 400 ou;
  - Se ele for divisível por 4 e não por 100.

```
algoritmo "ano-bissexto"
var
ano : inteiro
inicio
escreva("Digite um ano: ")
leia(ano)
se (ano % 400 = 0) ou ((ano % 4 = 0) e (ano % 100 <> 0)) entao
    escreva("É um ano bissexto!")
senao
    escreva("Não é um ano bissexto!")
fimse
finalgoritmo
```

# OPERADOR LÓGICO XOU

- Operador **OU EXCLUSIVO**
- O operador lógico '**xou**' difere do apenas num ponto específico
  - No 'ou', quando as duas expressões lógicas forem verdadeiras, o resultado é verdadeiro
  - No 'xou', quando as duas expressões lógicas forem verdadeiras, o resultado é falso



# TABELA VERDADE DO XOU

A	B	A xou B
F	F	F
F	V	V
V	F	V
V	V	F



# OPERADOR LÓGICO NÃO

- O operador lógico 'nao' nega uma expressão lógica
  - Ou seja, esse operador inverte o valor lógico
- Quando uma expressão lógica é verdadeira, e é usado o operador 'nao' sobre ela, a expressão lógica passa a ser falsa
- Quando uma expressão lógica é falsa, e é usado o operador 'nao' sobre ela, a expressão lógica passa a ser verdadeira





# TABELA VERDADE DO NÃO

A	nao A
V	F
F	V



# LISTA DE EXERCÍCIOS III

- Veja no site a Lista de Exercícios III



# REFERÊNCIAS

- Slides de aula do Prof. Abrahão Lopes
  - <http://docente.ifrn.edu.br/abrahaolopes>
- Slides de aula do Prof. Jalerson Lima
  - <http://www.jalersonlima.com.br>

