



# Estruturas de Decisão

---

Prof. MSc. Tiago Araújo

[tiagodavi70@gmail.com](mailto:tiagodavi70@gmail.com)

# Estruturas de Controle

---

As estruturas de entrada e saída são uma parte de estrutura de controle de fluxo, que permite que a linearidade da execução de um programa possa ser alterada.

Entre as estruturas de controle estão:

- Entrada e saída
- Estrutura de decisão
- Múltipla escolha
- Repetição

# Estruturas de Decisão

---

Estruturas de **controle de fluxo**, executando um ou vários comandos **se a condição** testada **for verdadeira** e, em alguns casos, executando um ou vários comandos se for **falsa** (LOPES e GARCIA, 2002).

Condição é uma comparação que possui dois valores possíveis, verdadeiro ou falso (ASCENCIO e CAMPOS, 2008).

# Estruturas de Decisão

---

Estruturar de forma automática essas decisões é fundamental para criar algoritmos simples

Operações Relacionais e Operações lógicas são a base para controle de fluxo em um algoritmo

# Operadores Relacionais

---

Operador	Ação
>	Maior que
>=	Maior ou igual a
<	Menor que
<=	Menor ou igual a
=	Igual a
<>	Diferente de

$3 < 7$

$10 > 2$

$31 = 31$

$3 \leq 9$

$12 \geq 4$

$31 \neq 13$

# Fixação

---

1)  $(3 + 7) / 2 > 3 + 7 / 2$

2)  $5 * (3 * 4) = (5 * 3) * 4$

3)  $21 / 3 >= 63 / 9$

4)  $21 / 3 <= 63 / 9$

5)  $21 / 3 >= 63 / 8 + 1$

A precedência é como na aritmética: primeiro os parênteses e depois multiplicações e divisões, seguido de somas e subtrações, sempre da esquerda para direita

# Operações Lógicas

---

Na álgebra de Boole Existem apenas 2 estados lógicos:

- Sim / Verdadeiro / 1
- Não / Falso / 0

Os operadores dessa álgebra estão implementados em linguagens de programação e diretamente em dispositivos físicos: **E**, **OU** e **NÃO**:

# Tabela Verdade E

---

Condição	Resultado
Verdadeiro <b>E</b> Verdadeiro	Verdadeiro
Verdadeiro <b>E</b> Falso	Falso
Falso <b>E</b> Verdadeiro	Falso
Falso <b>E</b> Falso	Falso



# Fixação

---

Quais as respostas para as questões abaixo, considerando:

$X \leftarrow 5$   
 $Y \leftarrow 10$

- 1)  $X = 5$  E  $Y = 11$
- 2)  $X \neq 2$  E  $Y \neq 11$
- 3)  $X = 5$  E  $Y < 10$
- 4)  $X \geq 3$  E  $Y \leq 12$

# Tabela Verdade OU

---

Condição	Resultado
Verdadeiro <b>OU</b> Verdadeiro	Verdadeiro
Verdadeiro <b>OU</b> Falso	Verdadeiro
Falso <b>OU</b> Verdadeiro	Verdadeiro
Falso <b>OU</b> Falso	Falso

# Fixação

---

Quais as respostas para as questões abaixo, considerando:

$X < -5$   
 $Y < -10$

- 1)  $X = 5$  OU  $Y = 11$ ;
- 2)  $X < > 2$  OU  $Y < > 11$ ;
- 3)  $X < > 5$  OU  $Y < 10$ ;
- 4)  $X \geq 3$  OU  $Y \leq 12$

# Tabela Verdade NÃO

---

	Condição	Resultado
<b>NÃO</b>	<i>Verdadeiro</i>	Falso
<b>NÃO</b>	<i>Falso</i>	Verdadeiro

# Fixação

---

Quais as respostas para as questões abaixo, considerando:

$X < 5$

$Y < 10$

- 1) NÃO ( $X = 5$ )
- 2) NÃO ( $X = Y$ )
- 3) NÃO ( $Y > 10$ )
- 4) NÃO ( $Y \leq 10$ )

# Fixação

---

Considere:

var

x, y : inteiro

...

x <- 15

y <- 23

Qual a resposta para cada expressão abaixo?

A) NÃO (x = 10) E (y > 20)

B) NÃO (x = 15) OU NÃO (y <> 23)

C) ((y - x) < y) E NÃO((x + y) > 40)

# Estruturas de Decisão

---

Executam uma sequência de comandos de acordo com o resultado de um teste

- **Decisão Simples:** seleciona ou ignora uma única ação
- **Decisão Composta:** seleciona entre duas ações ou grupos de ações diferentes

# Estruturas de Decisão - VisuAlg

---

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

```
se <expressão-lógica> entao  
    <sequência-de-comandos-1>  
senao  
    <sequência-de-comandos-2>  
fimse
```

```
escolha <expressão-de-seleção>  
caso <exp11>, <exp12>, ..., <exp1n>  
    <sequência-de-comandos-1>  
caso <exp21>, <exp22>, ..., <exp2n>  
    <sequência-de-comandos-2>  
outrocaso  
    <sequência-de-comandos-extra>  
fimescolha
```



# Exemplo

Algoritmo "decisao\_simples"

Var

a: inteiro

Inicio

a ← 5

se a > 2 então

    escreval("a é maior que 2")

fimse

se a > 8 então

    escreval("a é maior que 8")

senão

    escreval("a é menor que 8")

fimse

Fimalgoritmo

# Exemplo

Algoritmo "decisao\_maior"

Var

num1, num2: inteiro

Inicio

escreva("Numero 1: ")

leia(num1)

escreva("Numero 2: ")

leia(num2)

se num1 > num2 entao

    escreval(num1, " é maior")

fimse

se num1 < num2 entao

    escreval(num2, " é maior")

fimse

Fimalgoritmo

# Exemplo

```
Algoritmo "decisao_avaliacao"
```

```
var
```

```
num1, num2, num3: inteiro
```

```
resultado: real
```

```
Inicio
```

```
leia(num1)
```

```
leia(num2)
```

```
leia(num3)
```

```
resultado <- (num1 + num2 + num3)/3;
```

```
Se resultado >= 5 Entao
```

```
    escreva("Aprovado")
```

```
fimse
```

```
Se resultado < 5 Entao
```

```
    escreva("Reprovado")
```

```
fimse
```

```
Fimalgoritmo
```

# Exemplo

```
Algoritmo "decisao_times"
```

```
var  
time: caractere
```

```
inicio
```

```
escreva("Entre com o nome de um time de futebol: ")
```

```
leia(time)
```

```
// Relações no escolha-caso só funcionam com maiúsculas
```

```
escolha time
```

```
caso "FLAMENGO", "FLUMINENSE", "VASCO", "BOTAFOGO"
```

```
    escreval("É um time carioca.")
```

```
caso "SÃO PAULO", "PALMEIRAS", "SANTOS", "CORÍNTHIANS"
```

```
    escreval("É um time paulista.")
```

```
outrocaso
```

```
    escreval("É de outro estado.")
```

```
fimescolha
```

```
fimalgoritmo
```

# Prática

---

Corrija os erros:

a) Se `c < 7` então

`escreva("c é menor que 7")`

`fimse`

b) Se `c <- 7`

`então escreva("c é igual a 7")`

`fimse`

c) Se `c => 7` então

`escreva("c é maior ou igual a 7")`

d) se `c <> 7`

`então escreva("c é diferente de 7")`

`fimse`

# Prática

---

Faça um algoritmo que leia um número N e imprima "F1", "F2" ou "F3", conforme a condição:

- "F1", se  $N < 10$ ;
- "F2", se  $N = 10$ ;
- "F3", se  $N > 10$

Escreva um algoritmo que, para uma conta bancária, leia o seu número, o saldo, o tipo de operação a ser realizada (depósito ou retirada) e o valor da operação. Após, determine e mostre o novo saldo.

Faça um algoritmo que leia um inteiro e mostre uma mensagem indicando se este número é par ou ímpar.