

# Técnicas de Desenvolvimento de Algoritmos



Cruzeiro do Sul Virtual  
Educação a distância



# Material Teórico



Estruturas de Decisão

**Responsável pelo Conteúdo:**

Prof.<sup>a</sup> Me. Ana Fernanda Gomes Ascencio

**Revisão Textual:**

Jaquelina Kutsunugi

**Revisão Técnica:**

Prof.<sup>a</sup> Esp. Margarete Eliane da Silva Almendro



# UNIDADE

## Estruturas de Decisão



- Estruturas Condicionais em Algoritmos.



### OBJETIVO DE APRENDIZADO

- Desenvolver algoritmos em fluxograma e em pseudocódigo com estruturas condicionais.





# Orientações de estudo

Para que o conteúdo desta Disciplina seja bem aproveitado e haja maior aplicabilidade na sua formação acadêmica e atuação profissional, siga algumas recomendações básicas:



Determine um horário fixo para estudar.

Procure manter contato com seus colegas e tutores para trocar ideias! Isso amplia a aprendizagem.

Aproveite as indicações de Material Complementar.

Conserve seu material e local de estudos sempre organizados.

Seja original!  
Nunca plagie trabalhos.

Assim:

- ✓ Organize seus estudos de maneira que passem a fazer parte da sua rotina. Por exemplo, você poderá determinar um dia e horário fixos como seu “momento do estudo”;
- ✓ Procure se alimentar e se hidratar quando for estudar; lembre-se de que uma alimentação saudável pode proporcionar melhor aproveitamento do estudo;
- ✓ No material de cada Unidade, há leituras indicadas e, entre elas, artigos científicos, livros, vídeos e sites para aprofundar os conhecimentos adquiridos ao longo da Unidade. Além disso, você também encontrará sugestões de conteúdo extra no item **Material Complementar**, que ampliarão sua interpretação e auxiliarão no pleno entendimento dos temas abordados;
- ✓ Após o contato com o conteúdo proposto, participe dos debates mediados em fóruns de discussão, pois irão auxiliar a verificar o quanto você absorveu de conhecimento, além de propiciar o contato com seus colegas e tutores, o que se apresenta como rico espaço de troca de ideias e de aprendizagem.



Não se esqueça de se alimentar e de se manter hidratado.



# Estruturas Condicionais em Algoritmos

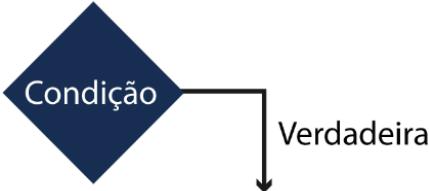
Nesta Unidade, vamos continuar conhecendo um pouco mais sobre as estruturas que compõem os algoritmos, salientando que a implementação dos algoritmos, utilizando qualquer Linguagem de Programação, permite seu teste efetivo. Neste curso, os testes serão realizados na ferramenta VisualG.

Na Unidade I, abordamos as definições de algoritmos e as três técnicas mais utilizadas para o seu desenvolvimento. Na Unidade II, abordados alguns algoritmos que utilizam apenas estrutura sequencial, ou seja, entrada dos dados, processamento e saída. A partir desta Unidade, vamos abordar algoritmos que apresentam desvios em seus comandos. Os desvios são denominados Estruturas Condicionais.

As estruturas condicionais estão divididas em: Estruturas Condicionais Simples, Compostas e de Caso.

A primeira estrutura condicional a ser abordada é a estrutura condicional simples, em que o comando ou os comandos só serão executados se a condição for verdadeira. Uma condição é uma comparação que possui dois valores possíveis, verdadeiro ou falso.

Quadro 1 - Estrutura condicional simples em fluxograma e em pseudocódigo

fluxograma	pseudocódigo
	<b>se (condição) então</b> <b>    comando</b> <b>fimse</b>

As condições são comparações que resultam em verdadeiro ou falso. Cada condição terá um operador relacional e estes estão descritos a seguir:

Quadro 2 - Operadores relacionais utilizados nas condições

Operador	Função
=	igual
<>	diferente
<=	menor igual
>=	maior igual
<	menor
>	maior

Quando é necessário ter mais de uma condição, é necessária a utilização de operadores lógicos e estes são: E, OU e NÃO, sendo usados para conjunção, disjunção e negação respectivamente.

Os resultados dos operadores lógicos estão descritos a seguir:

Quadro 3 - Operadores lógicos utilizados nas condições compostas

TABELA E	TABELA OU	TABELA NÃO
V e V = V	V ou V = V	Não V = F
V e F = F	V ou F = V	Não F = V
F e V = F	F ou V = V	
F e F = F	F ou F = F	



Para os valores  $x = 2$  e  $y = 3$ , a condição abaixo é verdadeira?

```
se (x > 2 E y <= 3) entao
    escreva ("VERDADEIRA")
senao
    escreva ("FALSA")
```

Neste caso a mensagem a ser exibida é a palavra FALSA.

A segunda estrutura condicional a ser abordada é a estrutura condicional composta, em que o comando1 será executado se a condição for verdadeira e o comando 2 será executado se a condição for falsa.

Quadro 4 - Estrutura condicional composta em fluxograma e em pseudocódigo

fluxograma	pseudocódigo
	<pre>se(condição) entao     comando1 senao     comando2 fimse</pre>

A terceira e última estrutura de decisão é denominada CASO e está ilustrada a seguir:

Quadro 5 - Estrutura de caso em fluxograma e em pseudocódigo

fluxograma	pseudocódigo
	ESCOLHA operacao CASO “1” comando1 CASO “2” comando2 CASO “3” comando3 CASO “N” comandoN FIMESCOLHA fimse

Fonte: elaborado pela autora.



Para quais valores de num e cont, a condição abaixo é verdadeira?

Desse ponto em diante, vamos fazer alguns exemplos de algoritmos que utilizam estruturas condicionais.

**EXEMPLO 1** - Faça um algoritmo que recebe duas notas, calcule a média aritmética e mostre mensagem de Aprovado ou Reprovado, considerando para aprovação média de, no mínimo, 7.

Abaixo, encontra-se o desenvolvimento do ALGORITMO 1 em fluxograma:

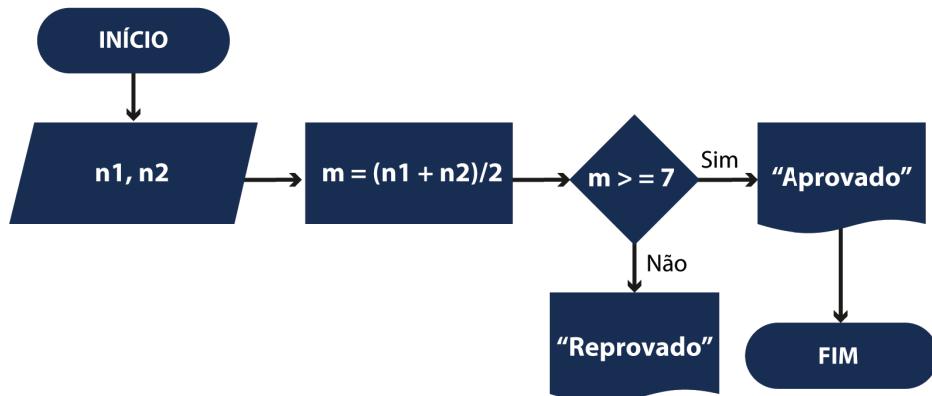


Figura 1 - Fluxograma do algoritmo exemplo 1

Primeira forma de resolução:

```

algoritmo "exemplola"
// Função: calcular a média aritmética entre duas notas
// mostrar mensagem de aprovado ou reprovado
// Autor: Ana Fernanda /*revisão Margarete E. S. Almendro
// Data: 24/12/2018      /*07/02/2020
// RESOLUÇÃO UTILIZANDO ESTRUTURAS CONDICIONAIS SIMPLES
// seção de declarações de variáveis e constantes
var m, n1, n2: real
inicio
    escreval("Digite duas notas")
    leia(n1,n2)
    m <- (n1+n2)/2
    se m >= 7 entao
        escreval("Aprovado")
    fimse
    se m < 7 entao
        escreval ("Reprovado")
    fimse
fimalgoritmo

```

Segunda forma de resolução:

```

algoritmo "exemplo1B"
// Função: calcular a média aritmética entre duas notas
// mostrar mensagem de aprovado ou reprovado
// Autor: Ana Fernanda
// Data: 27/12/2018
// RESOLUÇÃO UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA
// seção de declarações de variáveis e constantes

```

```

var m, n1, n2: real
inicio
    escreval("Digite duas notas")
    leia(n1,n2)
    m <- (n1+n2)/2
    se m >= 7 entao
        escreval ("Aprovado")
    senao
        escreval ("Reprovado")
    fimse
fimalgoritmo

```

**EXEMPLO 2** - Faça um programa que receba três notas de um aluno, calcule e mostre a média aritmética e a mensagem constante no quadro a seguir. Aos alunos que ficaram para exame, calcule e mostre a nota que deverão tirar para serem aprovados, considerando que a média exigida é 6,0 (ASCENCIO; CAMPOS, 2012, p. 62).

Quadro 6 - Regras para a emissão de mensagem do exemplo 2

Média aritmética	Mensagem
0  ----- 3	Reprovado
3  ----- 7	Exame
7  -----  10	Aprovado

Abaixo, encontra-se o desenvolvimento do ALGORITMO 2 em fluxograma:

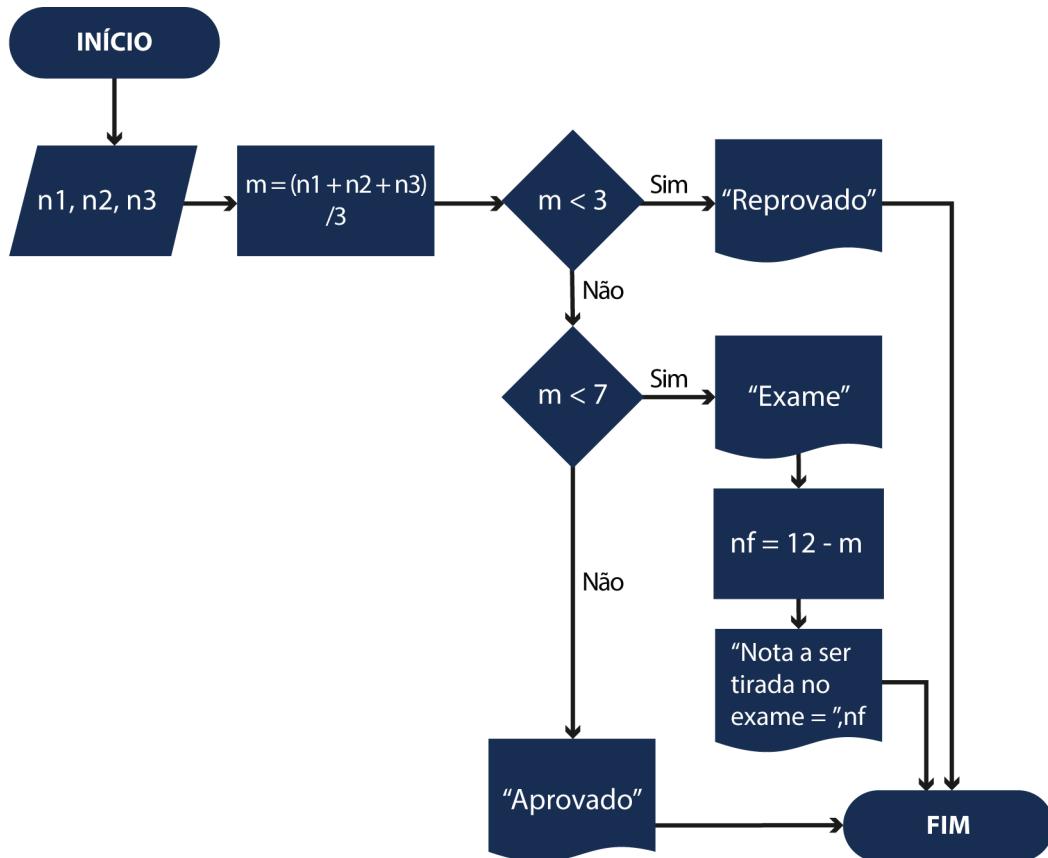


Figura 2 - Fluxograma do algoritmo exemplo 2

algoritmo “exemplo2a”

```

// Função: calcular a média aritmética entre três notas
// mostrar mensagem de aprovado, exame ou reprovado
// No caso de exame mostrar a nota a ser tirada para aprovação
// Autor: Ana Fernanda /*revisão Margarete E. S. Almendro
// Data: 24/12/2018      /*07/02/2020
  
```

```
// RESOLUÇÃO UTILIZANDO ESTRUTURAS CONDICIONAIS SIMPLES
// seção de declarações de variáveis e constantes
var m, n1, n2, n3, nf: real
inicio
    escreval("Digite três notas")
    leia(n1,n2,n3)
    m <- (n1+n2+n3) / 3
    escreval("Média = ",m)
    se (m >= 0) E (m < 3) entao
        escreval("Reprovado")
    fimse
    se (m >= 3) E (m < 7) então
        escreval ("Exame")
        nf <- 10 - m
        escreval("Nota a ser tirada no exame = ",nf)
    fimse
    se (m >= 7) então
        escreval("Aprovado")
    fimse
fimalgoritmo
```

Outro exemplo de construção com o mesmo propósito.

algoritmo "exemplo2b"

```
// Função: calcular a média aritmética entre três notas
// mostrar mensagem de aprovado, exame ou reprovado
// No caso de exame mostrar a nota a ser tirada para aprovação
// Autor: Ana Fernanda /*revisão Margarete E. S. Almendro
// Data: 24/12/2018      /*07/02/2020
// RESOLUÇÃO UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA
// seção de declarações de variáveis e constantes
```

```

var m, n1, n2, n3, nf: real
inicio
    escreval("Digite três notas")
    leia(n1,n2,n3)
    m <- (n1+n2+n3)/3
    escreval("Média =",m)
    se (m < 3) entao
        escreval("Reprovado")
    senao
        se (m < 7) entao
            escreval ("Exame")
            nf <- 10 - m
            escreval("Nota a ser tirada no exame =",nf)
        senao
            escreval("Aprovado")
        fimse
    fimse
fimalgoritmo

```



Qual o erro de lógica do algoritmo abaixo?

algoritmo "reflita"

```

// Função: calcular a média aritmética entre três notas
// mostrar mensagem de aprovado, exame ou reprovado
// Autor: Ana Fernanda /*revisão Margarete E. S. Almendro
// Data: 24/12/2018      /*07/02/2020
// seção de declarações de variáveis e constantes
var m, n1, n2, n3: real
inicio

```

```

escreval("Digite três notas")
leia(n1,n2,n3)
m <- (n1+n2+n3)/3
escreval("Média = ",m)
se (m < 3) entao
    escreval("Reprovado")
fimse
se (m >= 7) entao
    escreval ("Aprovado")
senao
    escreval ("Exame")
fimse
fimalgoritmo

```

**EXEMPLO 3** - Faça um algoritmo que receba dois números e mostre o maior. Caso os números sejam iguais, mostrar mensagem.

Abaixo, encontra-se o desenvolvimento do ALGORITMO 3 em fluxograma:

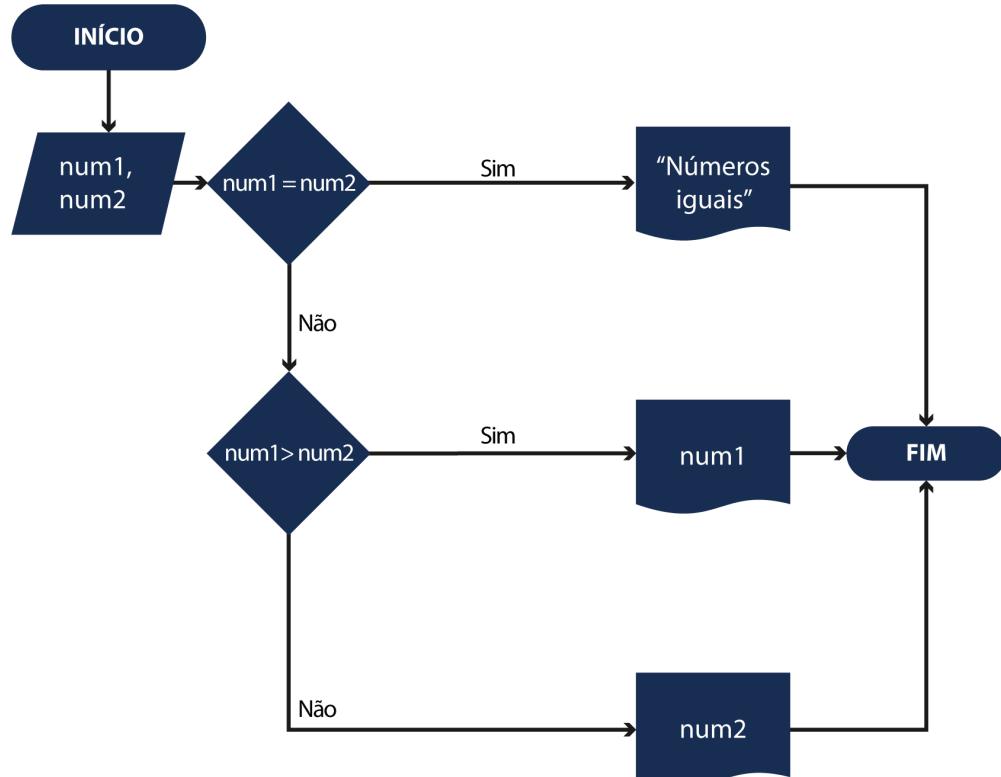


Figura 3 - Fluxograma do algoritmo exemplo 3

```

algoritmo "algoritmo3a"
// Função: receber dois números e mostrar o maior
// Autor: Ana Fernanda /*revisão Margarete E. S. Almendro
// Data: 24/12/2018      /*07/02/2020
// RESOLUÇÃO UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES
// seção de declarações de variáveis e constantes
var num1, num2: real
inicio
    escreval("Digite o primeiro número:")
    leia(num1)
    escreval("Digite o segundo número:")
    leia(num2)
    se (num1 = num2) entao
        escreval("Número iguais")
    fimse
    se (num1 > num2) entao
        escreval ("O número ",num1," é o maior")
    fimse
    se (num1 < num2) entao
        escreval ("O número ",num2," é o maior")
    fimse
fimalgoritmo

```

A seguir uma nova solução para o mesmo enunciado.

```

algoritmo "algoritmo3b"
// Função: receber dois números e mostrar o maior
// Autor: Ana Fernanda /*revisão Margarete E. S. Almendro
// Data: 24/12/2018      /*07/02/2020
// RESOLUÇÃO UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA

```

```
// seção de declarações de variáveis e constantes  
var num1, num2: real  
  
inicio  
    escreval("Digite o primeiro número:")  
    leia(num1)  
    escreval("Digite o segundo número:")  
    leia(num2)  
    se (num1 = num2) entao  
        escreval("Números iguais")  
    senao  
        se (num1 > num2) entao  
            escreval ("O número ",num1," é o maior")  
        senao  
            escreval ("O número ",num2," é o maior")  
    fimse  
fimse  
fimalgoritmo
```



Se o usuário digitar os números 7 e 5, qual será a mensagem mostrada pelo algoritmo abaixo? Está correto?

```
algoritmo "reflita2"  
// Função: receber dois números e mostrar o maior  
// Autor: Ana Fernanda
```

```

// Data: 27/12/2018

// seção de declarações de variáveis e constantes

var num1, num2: real

inicio

    escreval("Digite o primeiro número:")

    leia(num1)

    escreval("Digite o segundo número:")

    leia(num2)

    se (num1 > num2) entao

        escreval ("O número „,num1,“ é o maior")

    fimse

    se (num1 < num2) entao

        escreval ("O número „,num2,“ é o maior")

    senao

        escreval("Número iguais")

    fimse

fimalgoritmo

```

0 número 7 é o maior. Número iguais.

**EXEMPLO 4** - Faça um algoritmo que receba os três lados de um triângulo e mostre sua classificação conforme regras abaixo:

- triângulo escaleno: três lados diferentes;
- triângulo isósceles: dois lados iguais;
- triângulo equilátero: três lados iguais.

A seguir, encontra-se o desenvolvimento do ALGORITMO 4 em fluxograma.

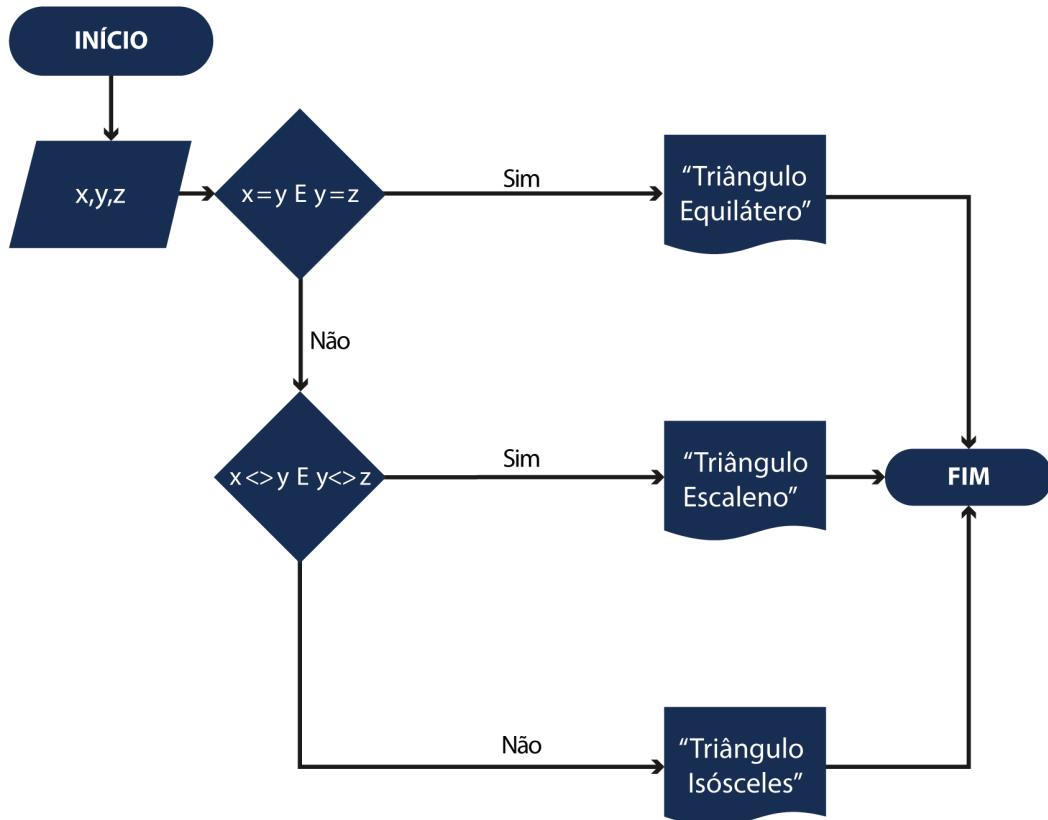


Figura 4 - Fluxograma do algoritmo exemplo 4

algoritmo "exemplo4a"

```

// Função: receber os três lados de um triângulo e classificar
// Autor: Ana Fernanda /*revisão Margarete E. S. Almendro
// Data: 24/12/2018      /*07/02/2020
// RESOLUÇÃO UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES
// seção de declarações de variáveis e constantes
var x, y, z: real
inicio
    escreva("Digite o primeiro lado: ")
    leia(x)
    escreva("Digite o segundo lado: ")
    leia(y)
    escreva("Digite o terceiro lado: ")
    leia(z)
  
```

```

    se (x = y) E (y = z) entao
        escreval ("Triângulo Equilátero")
    fimse

    se (x <> y) E (x <> z) E (y <> z) entao
        escreval ("Triângulo Escaleno")
    fimse

    se (x = y) E (y <> z) então
        escreval("Triângulo Isósceles")
    fimse

    se (x = z) E (z <> y) então
        escreval("Triângulo Isósceles")
    fimse

    se (y = z) E (z <> x) então
        escreval("Triângulo Isósceles")
    fimse

fimalgoritmo
  
```

```

algoritmo "exemplo4b"
    // Função: receber os três lados de um triângulo e clas-
    sificar
    // Autor: Ana Fernanda /*revisão Margarete E. S. Almendro
    // Data: 24/12/2018      /*07/02/2020
    // RESOLUÇÃO UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA
    // seção de declarações de variáveis e constantes
    var x, y, z: real
    inicio
        escreva("Digite o primeiro lado: ")
        leia(x)
        escreva("Digite o segundo lado: ")
        leia(y)
        escreva("Digite o terceiro lado: ")
        leia(z)
        se (x = y) E (y = z) entao
            escreval ("Triângulo Equilátero")
  
```

```
senao
  se (x <> y) E (x <> z) E (y <> z) entao
    escreval ("Triângulo Escaleno")
  senao
    escreval ("Triângulo Isósceles")
  fimse
fimse
fimalgoritmo
```

As estruturas condicionais permitem que o processamento de um algoritmo seja desviado de acordo com uma condição.

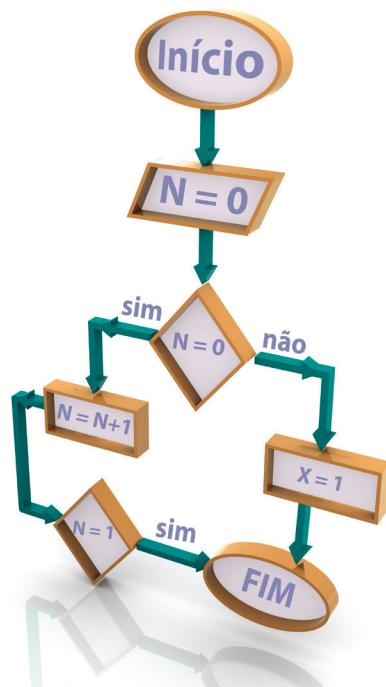


Figura 5 - Estruturas condicionais em algoritmos

Fonte: Nasirkhan / 123RF

Essas são algumas das estruturas condicionais em algoritmos apresentadas nesta Unidade.

# Material Complementar

## Indicações para saber mais sobre os assuntos abordados nesta Unidade:



### Livros

#### **Lógica de Programação com Pascal**

O livro “**Lógica de Programação com Pascal**”, no Capítulo 5, mostra a teoria sobre Estruturas Condicionais e vários exemplos. Ao final deste capítulo, há uma lista de exercícios a resolver.

ASCENCIO, A. F. G. **Lógica de Programação com Pascal**. São Paulo: Pearson, 1999.

#### **Fundamentos da Programação de Computadores**

O livro “**Fundamentos da Programação de Computadores**”, no Capítulo 4, mostra a teoria sobre Estruturas Condicionais e vários exemplos. Ao final deste capítulo, existem duas listas de exercícios, uma resolvida e a outra a resolver.

ASCENCIO, A. F. G.; CAMPOS, E. A. V. **Fundamentos da Programação de Computadores**. 3. ed. São Paulo: Pearson, 2012.

#### **Algoritmos. Teoria e Prática**

O livro “**Algoritmos. Teoria e Prática**”, do autor Thomas Cormen, é um clássico na área de Computação e aborda o desenvolvimento de algoritmos na teoria e na prática.

CORMEN, T. H. **Algoritmos. Teoria e Prática**. Rio de Janeiro: Campus, 2012.

#### **Algoritmos**

O livro “**Algoritmos**”, de José Augusto Manzano, é muito indicado para quem está começando a desenvolver algoritmos, tendo em vista a possível sequência didática de aprendizagem e uma linguagem apropriada para iniciantes.

MANZANO, J. A. N. G. **Algoritmos**. 28. ed. São Paulo: Pearson, 2016.

# Referências

ASCENCIO, A. F. G.; CAMPOS, E. A. V. **Fundamentos da Programação de Computadores**. 3. ed. São Paulo: Pearson, 2012.





**Cruzeiro do Sul**  
Educacional