

Lógica de Programação e Algoritmos

Profa. Eliane Oliveira Santiago

Estruturas de Decisão

Uma estrutura de decisão é um fluxo de controle utilizado para decidir qual o fluxo que o algoritmo seguirá. Uma estrutura de decisão determina qual conjunto de comandos ou bloco será executado após uma condição ser avaliada. Essa condição é representada por expressões lógicas e relacionais que podem ou não serem satisfeitas, isto é, podem retornar ao valor verdadeiro ou falso.

Uma estrutura de decisão pode ser simples, composta ou encadeada. Veremos a seguir como e quando trabalhar com cada uma delas.

Estrutura de Decisão Simples (se/então)

```
se (<condição>)  
    então  
        <comandos>;  
fimse;
```

Sintaxe

Exemplo

```
se (var < 0)  
    então  
        escreva("O número " , var , " é negativo");  
fimse;
```

Estrutura de Decisão Composta

(se ... então ... senão)

```
se (<condição>)  
    então  
        <comandos1>;  
    senão  
        <comandos2>;  
fimse;
```

Sintaxe

```
if (<condição>)  
    <comandos1>;  
else  
    <comandos2>;
```

```
se (numero < 0)  
    escreva("O número " , numero, " é negativo");  
senão  
    escreva("O número " , numero, " é não-negativo");  
fimse;
```

Equação de 2º. grau

Equação: $2x^2 + 3x + 5$  Instância do problema

Variáveis do problema: $ax^2 + bx + c$

$$\Delta = b^2 - 4.a.c$$

$$x = \frac{-b \pm \sqrt{\Delta}}{2.a}$$

Algoritmo Equação_de_2º_grau

Início

declare a, b, c , delta: inteiro

x1, x2 : real

//entrada

leia (a, b, c);

//processamento e saída

delta \leftarrow b*b - 4 * a * c;

se(delta \geq 0)

x1 \leftarrow (-1*b+raiz(delta))/(2*a)

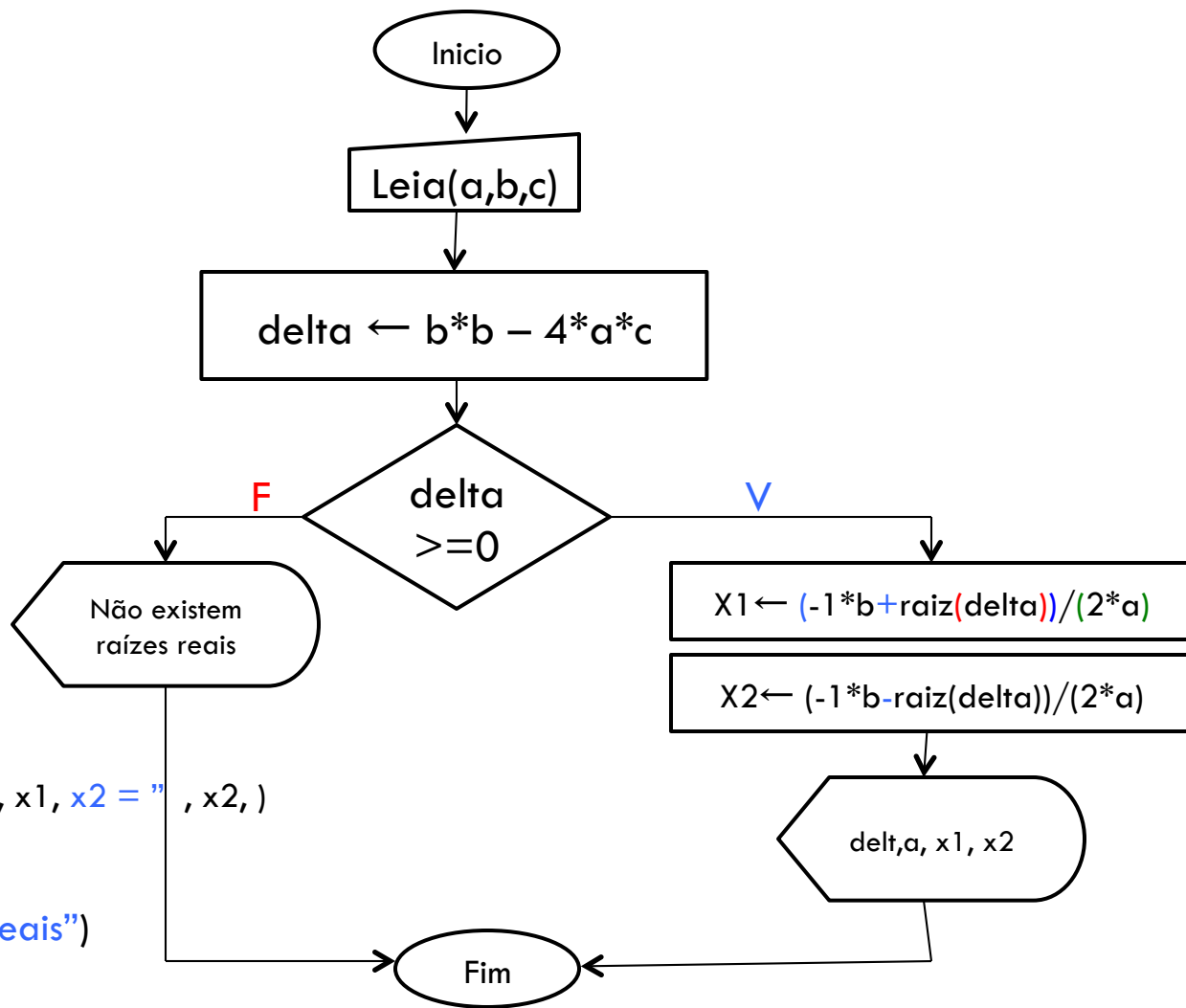
x2 \leftarrow (-1*b-raiz(delta))/(2*a)

escreva("Delta = ", delta, "x1 = ", x1, "x2 = ", x2,)

senão

escreva("Não existem raízes reais")

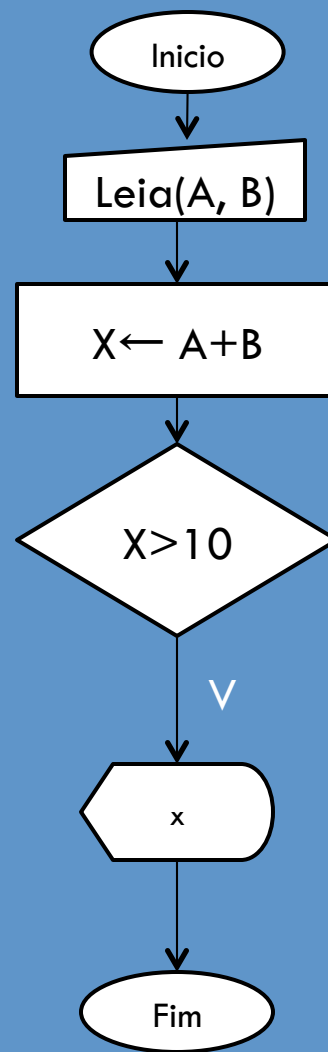
fim.



Exemplo 1.

Estrutura de decisão simples

```
programa ADIÇÃO_DE_NUMEROS_1  
var  
    A, B, X : real  
inicio  
    leia A,B  
    X ← A + B  
    se (X > 10) então  
        escreva X  
    fim_se  
fim
```



Estrutura de Decisão Encadeada

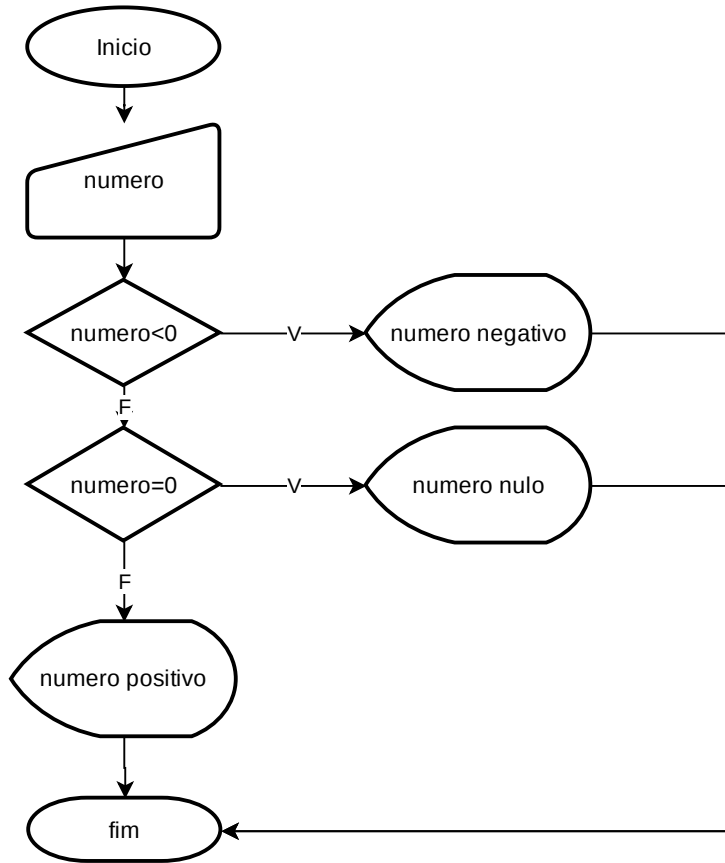
```
se (<condição1>)
  então
    se (<condição2>)
      então
        se (<condição3>)
          então
            ... // continua com outra estrutura de seleção ou coloca os comandos
          senão
            ... // continua com outra estrutura de seleção ou coloca os comandos
        fimse; // da condição 3
      fimse; // da condição 2
    senão
      se (<condição4>)
        então
          se (<condição5>)
            então
              ... // continua com outra estrutura de seleção ou coloca os comandos
            senão
              ... // continua com outra estrutura de seleção ou coloca os comandos
          fimse; // da condição 5
        fimse; // da condição 4
      fimse; // da condição 1
```


Exemplo 2. Estrutura de decisão encadeada

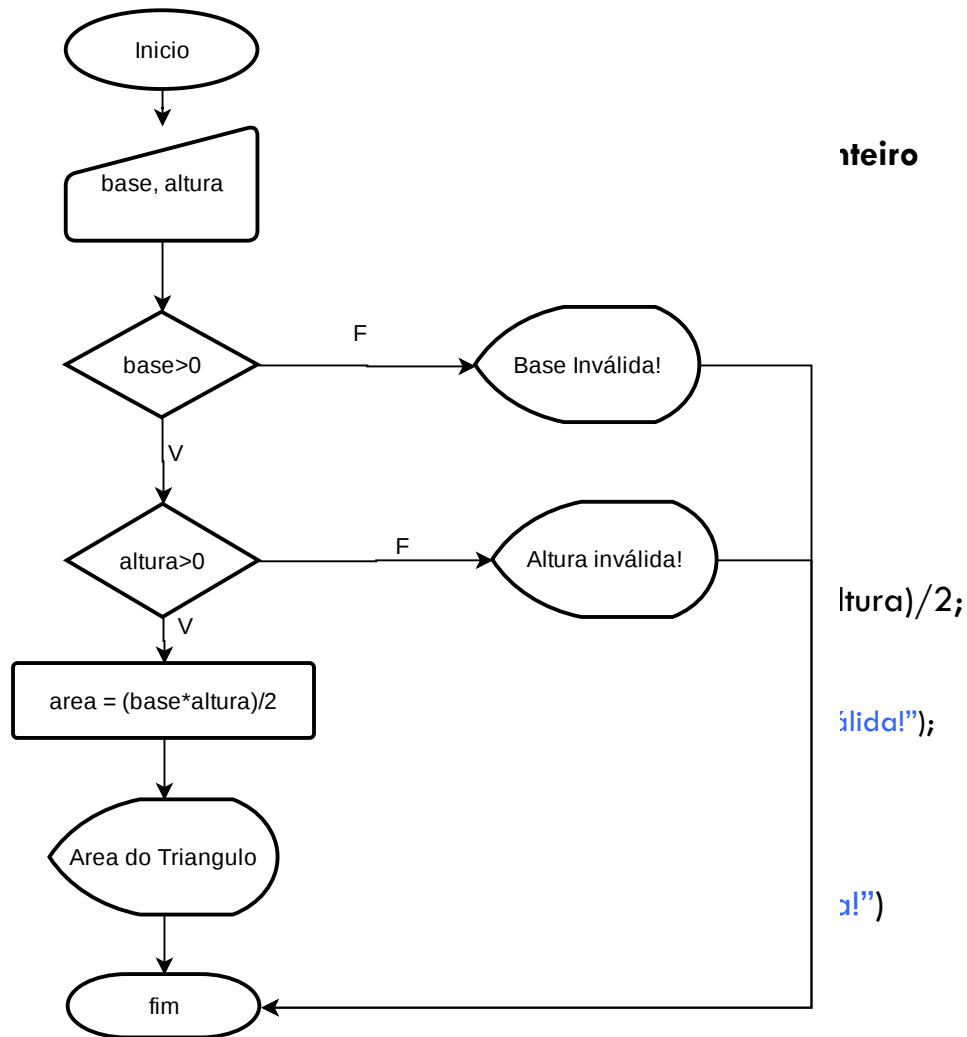
```
se (var < 0)
    então
        escreva("O número " , var , " é negativo");
    senão
        se (var = 0)
            então
                escreva("O número " , var , " é nulo");
            senão
                escreva("O número " , var , " é positivo");
        fimse;
    fimse;
```

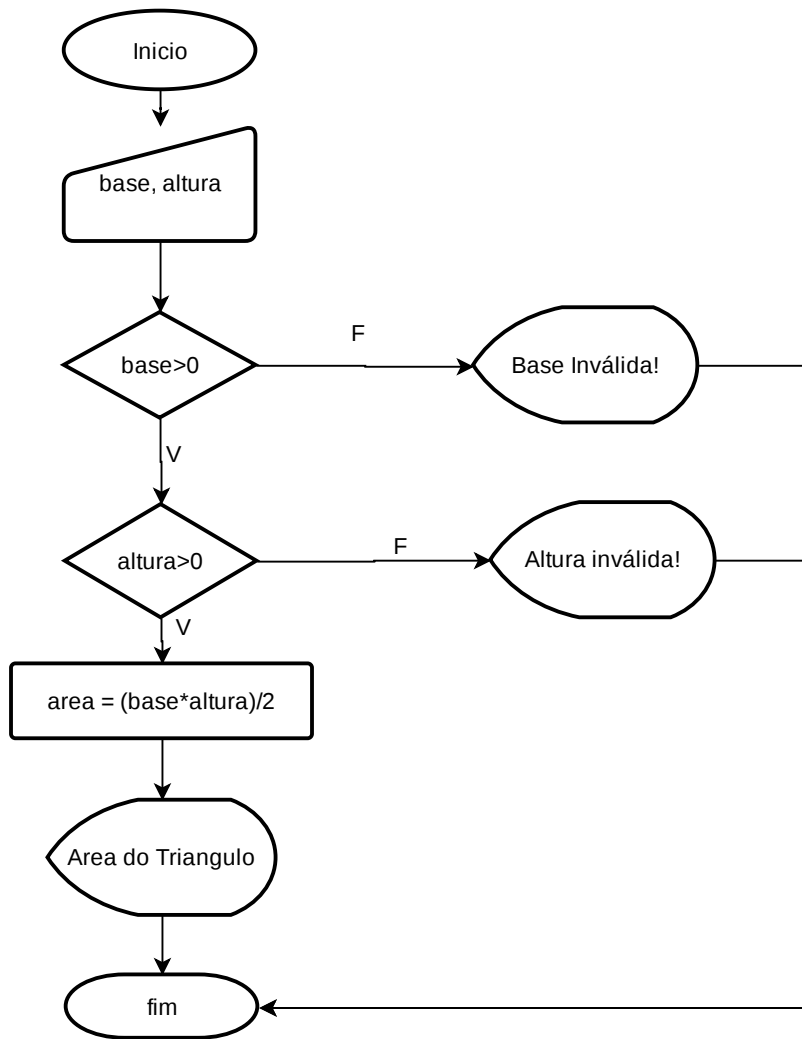
Exemplo 2. Estrutura de decisão encadeada

```
se (var < 0)
    escreva("O número " , var , " é negativo");
senão
    se (var = 0)
        escreva("O número " , var , " é nulo");
    senão
        escreva("O número " , var , " é positivo");
    fimse;
fimse;
```



Desenvolva um algoritmo que receba a base e a altura de um triângulo e verifique se os valores recebidos são válidos, ou seja, se são maiores que zero. Se os valores forem válidos, calcule e mostre a área desse triângulo, caso contrário, mostre uma mensagem informando que os valores são inválidos ao usuário.





Algoritmo ehTriangulo

Início

declare base, altura: **inteiro**

area : **real**

//entrada

leia (base, altura);

//processamento

se(base>0)

se(altura>0)

area ← (base*altura)/2;

senão

escreva("Altura inválida!");

fim_se

senão

escreva("Base inválida!");

fim_se

Fim.

Exercício Triângulo

Desenvolva um algoritmo que receba três valores numéricos inteiros, verifique e mostre se esses valores podem ser o comprimento dos lados de um triângulo.

Para ser triângulo, a soma de cada um dos lados deve ser menor que a soma dos outros dois.

Se falso, mostrar essa informação ao usuário, caso contrário, verificar se formam um triângulo equilátero, isósceles ou escaleno.

Triângulo equilátero tem os 3 lados iguais

Triângulo isósceles tem 2 lados iguais e 1 diferente

Triângulo escaleno tem os 3 lados diferentes.

Exercícios de estrutura de decisão simples, composta e encadeada (1)

1. Desenvolva um algoritmo que receba dois valores numérico inteiro, calcule e mostre o resultado da diferença do maior pelo menor valor.
2. Desenvolva um algoritmo que receba três valores numérico real e mostre-os em ordem crescente. Utilizar a estrutura de decisão encadeada.
3. Desenvolva um algoritmo que receba um valor inteiro, calcule e mostre o módulo desse número. O módulo de um número qualquer é esse número positivo, ou seja, o módulo de -5 é 5 e o módulo de 5 é 5 .
4. Desenvolva um algoritmo que receba dois valores numérico real, calcule e mostre o maior entre esses dois números.

Exercícios de estrutura de decisão

simples, composta e encadeada (2)

5. Desenvolva um algoritmo que receba os três coeficientes a , b e c de uma equação de segundo grau da forma $ax^2 + bx + c = 0$, verifique e mostre a existência de raízes reais e caso exista raíz(es) real(is), calcule e mostre essa(s) raíz(es).
6. Desenvolva um algoritmo que receba os valores referentes a quatro notas bimestrais de um aluno, calcule a média desse aluno e mostre a mensagem de aprovação (média ≥ 5) ou reprovação (média < 5) do aluno, juntamente com a média calculada.
7. Desenvolva um algoritmo que receba três valores obrigatoriamente em ordem crescente e um quarto valor não necessariamente nessa ordem. Esse algoritmo deve mostrar os quatro números em ordem decrescente.
8. Desenvolva um algoritmo que receba um valor numérico inteiro, verifique e mostre se esse valor é divisível por 2 e por 3.

Exercícios de estrutura de decisão simples, composta e encadeada (3)

9. Desenvolva um algoritmo que receba os valores da altura e do sexo de uma pessoa, calcule e mostre o peso ideal dessa pessoa, utilizando as seguintes regras:

- para homens: $(72.7 * \text{altura}) - 58$;
- para mulheres: $(62.1 * \text{altura}) - 44.7$.

Exercícios de estrutura de decisão simples, composta e encadeada (4)

10. Desenvolva um algoritmo que receba a data de nascimento de uma pessoa e a data atual, calcule e mostre se essa pessoa é maior de idade.

11. Desenvolva um algoritmo que receba três valores numérico real, calcule e mostre a média ponderada desses números, considerando que o primeiro número tem peso 2, o segundo número tem peso 3 e o terceiro número tem peso 5.

Exercícios de estrutura de decisão simples, composta e encadeada (5)

11) Desenvolva um algoritmo que receba o preço atual e a venda mensal média de um produto, calcule e mostre o novo preço, sabendo que:

- se a venda média mensal for menor que 500 e o preço atual menor que 30,00, então o produto sofre um aumento de 10%;
- se a venda média mensal for maior ou igual a 500 e menor que 1000 e o preço atual maior ou igual a 30,00 e menor que 80,00, então o produto sofre um aumento de 15%;
- se a venda mensal média for maior ou igual a 1000 e o preço atual maior ou igual a 80,00, então o produto sofre uma diminuição de 5%.

Exercícios de estrutura de decisão simples, composta e encadeada (6)

12. Desenvolva um algoritmo que receba o salário de um funcionário, calcule e mostre o valor do aumento salarial e o salário final a ser recebido pelo funcionário, considerando que se o funcionário:

- recebe menos que 500,00, terá reajuste de 100%;
- recebe mais ou igual a 500,00 e menos que 1.000,00, terá reajuste de 75%;
- recebe mais ou igual a 1.000,00 e menos que 1.500,00, terá reajuste de 50%;
- recebe mais ou igual a 1.500,00 e menos que 2.000,00, terá reajuste de 25%;
- recebe mais ou igual a 2.000,00 e menos que 3.000,00, terá reajuste de 10%;
- recebe mais ou igual a 3.000,00 e menos que 5.000,00, terá reajuste de 5%;
- recebe mais ou igual a 5.000,00, terá reajuste de 2%.

Bibliografias

BÁSICA

- GOMES, Ana Fernanda A. Campos, Edilene Aparecida V. Fundamentos da Programação de Computadores – Algoritmos, Pascal e C/C++. Prentice Hall, 2007.
- CARBONI, Irenice de Fátima. Lógica de Programação. Thomson.
- XAVIER, Gley Fabiano Cardoso. Lógica de Programação - Cd-rom. Senac São Paulo – 2007.

COMPLEMENTAR

- FORBELLONE, André Luiz Villar. Eberspache, Henri Frederico. Lógica de Programação – A construção de Algoritmos e Estrutura de Dados. Makron Books, 2005.
- LEITE, Mário - Técnicas de Programação – Brasport - 2006.
- PAIVA, Severino – Introdução à Programação – Ed. Ciência Moderna – 2008.
- PAULA, Everaldo Antonio de. SILVA, Camila Ceccatto da. Lógica de Programação –Viena – 2007.
- CARVALHO, Fábio Romeu, ABE, Jair Minoru. Tomadas de decisão com ferramentas da lógica paraconsistente anotada: Método Paraconsistente de Decisão (MPD), Editora Edgard Blucher Ltda. - 2012.