
Introdução a Algoritmos e Programas

Prof. Karl Apaza Agüero

Algoritmo

- ▶ É um processo sistemático para a resolução de um problema
- ▶ Computa uma saída (o resultado do problema) a partir de uma entrada (informações inicialmente conhecidas)



Algoritmo

► Exemplo

Algoritmo para somar dois números.

Entrada: Números $n1$ e $n2$.

Saída: Imprimir a soma.

Início

Obter os números $n1$ e $n2$.

Calcular soma: $n1 + n2$.

Imprimir soma.

Fim

Comentários

- ▶ Um algoritmo não representa, necessariamente, um programa de computador, senão os passos necessários para realizar uma tarefa
 - ▶ Um algoritmo é um método genérico que pode ser implementado em qualquer linguagem de programação
-

Algumas considerações para a construção de um algoritmo

- ▶ Análise do enunciado do problema
 - ▶ Identificação do objetivo
 - ▶ Identificação dos dados necessários para o funcionamento do algoritmo
 - ▶ Entrada de dados
 - ▶ Identificação das informações que devem ser apresentadas ao usuário como resultado do algoritmo
 - ▶ Saída de dados
 - ▶ Definição da sequência lógica de instruções em função das entradas e saídas
 - ▶ Processamento
-

Algumas considerações para a construção de um algoritmo

- ▶ Representação do algoritmo
 - ▶ Pseudocódigo
 - ▶ Fluxograma
 - ▶ Programa
 - ▶ Realização de testes de validação da solução proposta
 - ▶ Fornecer todos os possíveis dados de entrada para validar o algoritmo
-

Construção de um algoritmo

► Exemplo:

- Problema: Calcular a média de 3 notas de um aluno, imprimir a média e se ele foi aprovado ou não.
- Entradas?
- Processamento?
- Saídas?



Algoritmo

► Exemplo

Algoritmo para calcular a média de 3 notas de um aluno.

Entrada: Notas n_1 , n_2 e n_3 .

Saída: Média do aluno e se ele foi aprovado ou não.

Início

Obter as notas n_1 , n_2 e n_3 .

Calcular média. Usar a fórmula $(n_1 + n_2 + n_3) / 3.0$.

Imprimir a média.

Se a média ≥ 5.0 imprimir que o aluno foi aprovado.

Caso contrário imprimir que o aluno foi reprovado.

Fim

Algumas classificações de algoritmos

- ▶ Por paradigma
 - Estruturado: sequência, decisão e iteração
 - Orientado a objetos

 - ▶ Por campo de estudo
 - Algoritmos de busca, de ordenação, etc.

 - ▶ Por complexidade (análise do algoritmo)
 - Tempo constante, linear, exponencial, etc.
-

Comentários

- ▶ Qualidade de um algoritmo
 - ▶ Técnicas de Projeto de Algoritmos
 - ▶ Força Bruta, Divisão e Conquista
 - ▶ Recursos necessários
 - ▶ O tempo de execução e o espaço de armazenamento de dados
 - ▶ Complexidade computacional – Análises
 - ▶ Estrutura de dados usada
-

Representação de Algoritmos

- ▶ Ferramentas de Representação
 - ▶ Pseudocódigo
 - ▶ Fluxograma
 - ▶ Linguagem de programação
-

Pseudocódigo

- ▶ Forma genérica de se escrever um algoritmo
 - ▶ Usa-se uma linguagem simples, nativa a quem o escreve, de forma a ser entendida por qualquer pessoa (de um mesmo idioma)
 - ▶ Não é necessário o conhecimento da sintaxe de uma linguagem de programação específica
 - ▶ Não pode ser executado em um computador
-

Pseudocódigo

- Exemplo: Raízes de uma equação do 2º grau

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$
$$\Delta = b^2 - 4ac$$

Pseudocódigo

► Exemplo: Raízes de uma equação do 2º grau

```
1. Início
2.  Leia a, b, c
3.  Se a = 0 Então
4.    Escreva "A equação não é do 2º grau"
5.  Senão
6.    Calcule delta = b²-4ac
7.    Se delta > 0 Então
8.      Calcule x1 e x2
9.      Escreva "Raízes x1 e x2 Reais e distintas"
10.   Senão
11.     Se delta = 0 Então
12.       Calcule x1 = x2
13.       Escreva "Raízes x1 e x2 Reais e iguais"
14.     Senão
15.       Escreva "Não existem raízes Reais"
16. Fim
```

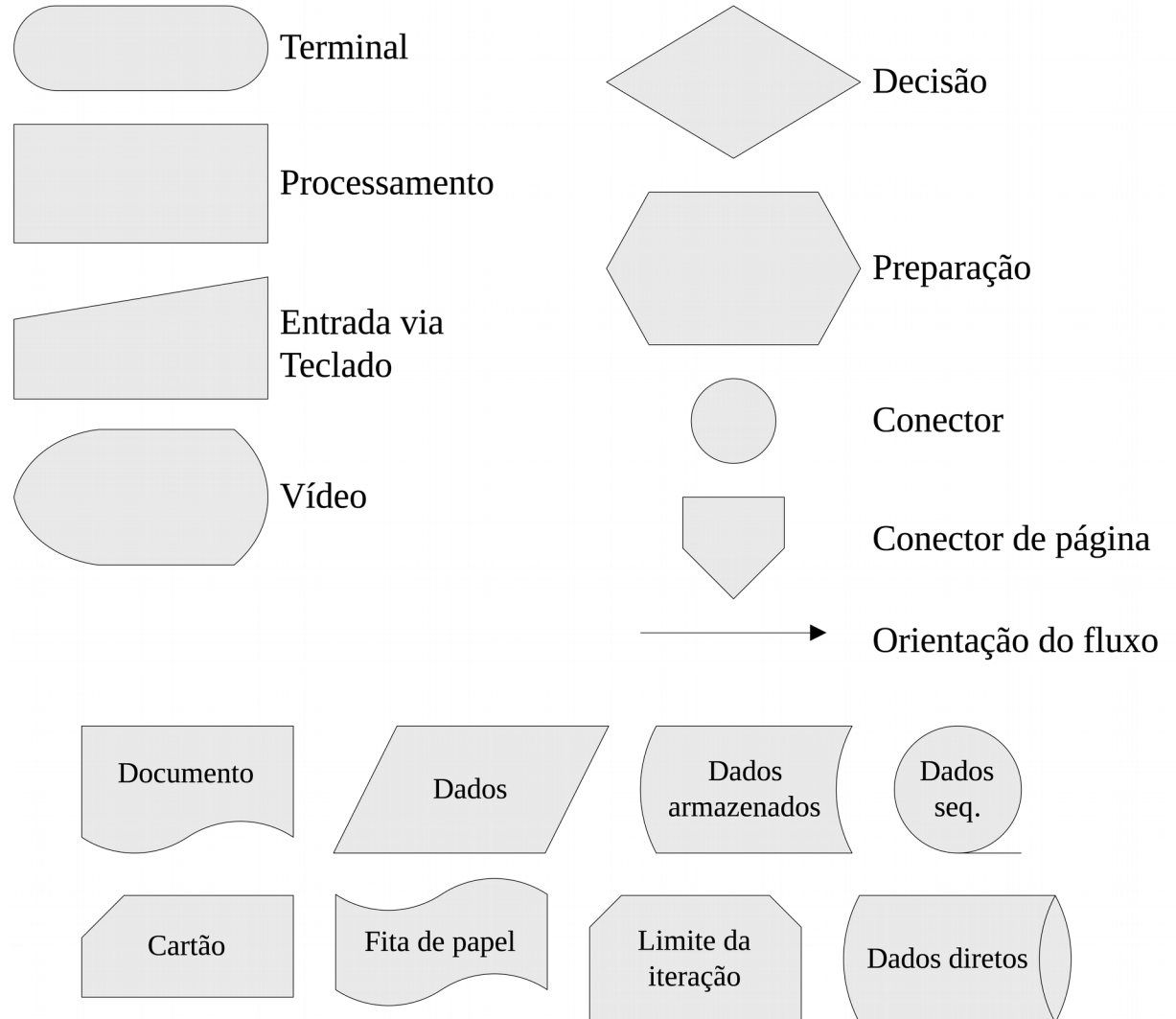
$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$
$$\Delta = b^2 - 4ac$$

Fluxograma

- ▶ Uso de símbolos gráficos
- ▶ Associação com as instruções a serem realizadas
- ▶ Recomendação:
 - ▶ Casos específicos e pouco extensos

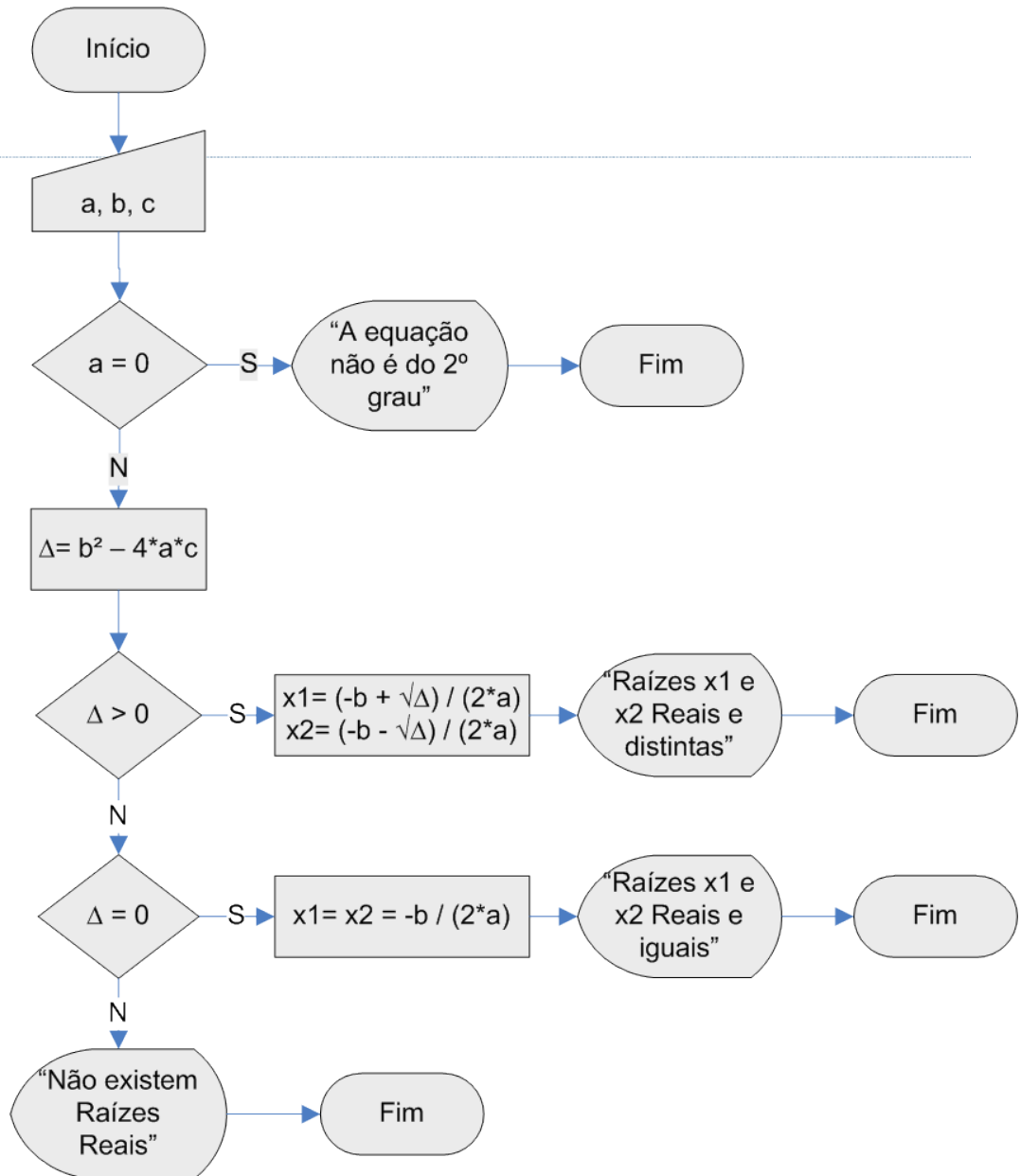
Fluxograma

► Simbologia



Fluxograma

- Exemplo:
Raízes de uma equação do 2º grau



Linguagem de Programação

- ▶ Definição

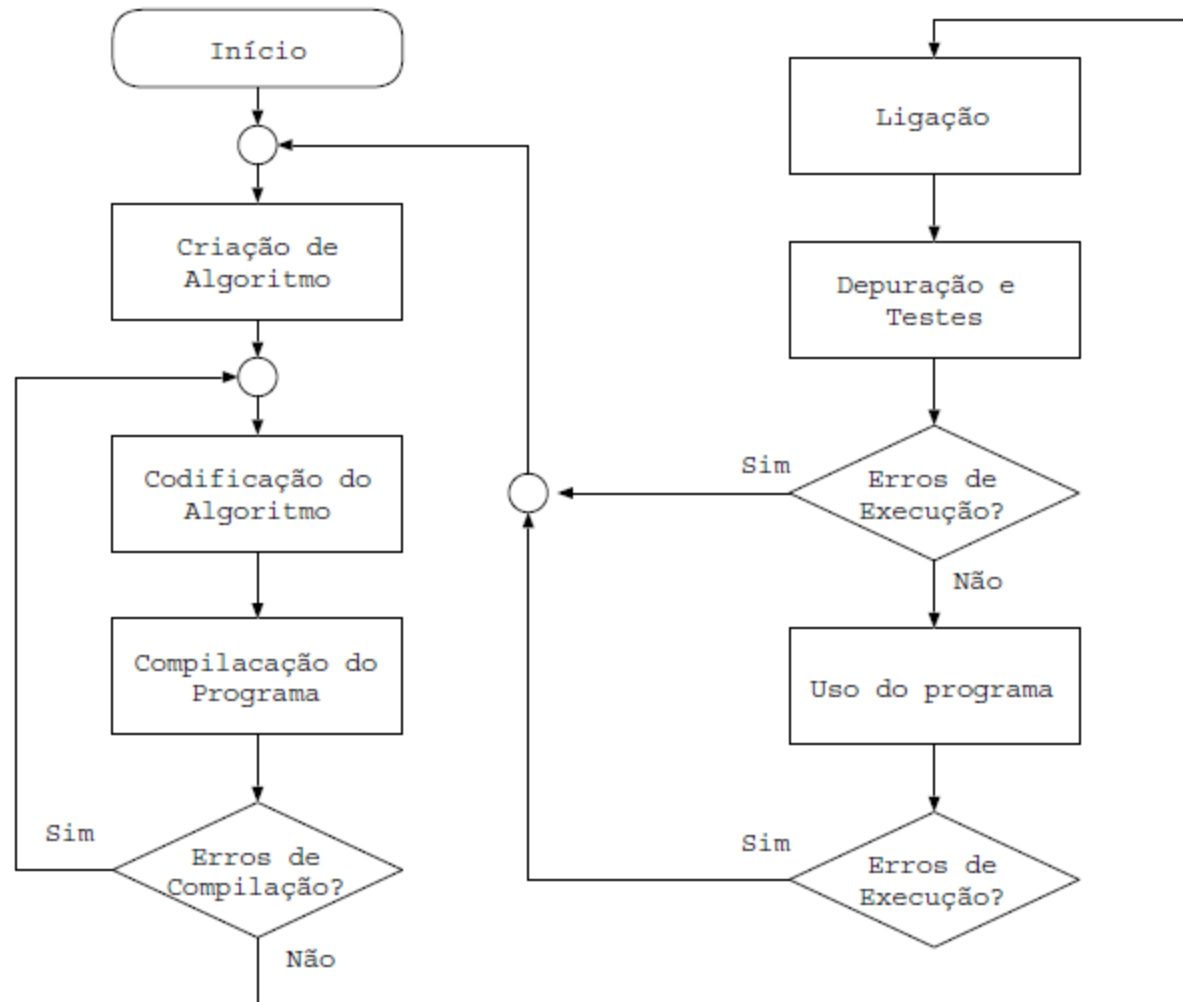
- ▶ É um conjunto de símbolos e regras sintáticas e semânticas usadas para expressar instruções para um computador

- ▶ Fase de codificação

- ▶ Transformação da solução construída (i. e. pseudocódigo, fluxograma etc) para uma linguagem de programação
-

Linguagem de Programação

► Ciclo de desenvolvimento de um programa



Linguagem de Programação

- ▶ Tempo de compilação
 - ▶ Refere-se aos eventos que acontecem durante o processo de compilação
 - ▶ Tempo de execução
 - ▶ Refere-se aos eventos que ocorrem enquanto o programa está sendo executado
 - ▶ Constantemente esses termos estão relacionados a mensagens de erros
 - ▶ Erros de tempo de compilação
 - ▶ Erros de tempo de execução
-

Linguagem de Programação C

- ▶ A linguagem C

- ▶ Criada em 1972, por Dennis Ritchie, no AT&T Bell Labs
 - ▶ C é uma das linguagens de programação mais populares
 - ▶ De propósito geral
 - ▶ Compilada
 - ▶ Estruturada
-

Linguagem de Programação C

- ▶ Exemplo: Raízes de uma equação do 2º grau

```
#include <stdio.h>
#include <math.h>
int main() {
    float a, b, c, delta, x1, x2;
    printf("Valor de a: ");
    scanf("%f",&a);
    printf("Valor de b: ");
    scanf("%f",&b);
    printf("Valor de c: ");
    scanf("%f",&c);

    if (a == 0) {
        printf("A equacao nao e do 2o grau\n");
    } else {
        delta = b*b-4*a*c;
        if (delta > 0) {
            float raiz = sqrtf(delta);
            x1 = (a*a + raiz)/2*a;
            x2 = (a*a - raiz)/2*a;
            printf("Raízes x1 e x2 Reais e Distintas:\n x1=%f\n x2=%f\n", x1, x2);
        } else if (delta == 0) {
            x1 = x2 = a/2;
            printf("Raizes x1 e x2 Reais e Iguais:\n x1=x2=%f\n", x1);
        } else {
            printf("Nao Existem Raizes Reais!\n");
        }
    }
    return 0;
}
```