

Algoritmos e Programação

(UFSM00013)

Prof. Dr. Joaquim V. C. Assunção

CENTRO DE TECNOLOGIA
UFSM
2025



Definição

“Sequencia finita de instruções ou operações básicas cuja execução, em tempo finito, resolve um problema computacional, qualquer que seja sua instância” Salvetti (98).

Lógica \rightarrow algoritmo

- Um algoritmo tem por objetivo representar a essência do raciocínio envolvido na Lógica de Programação, permitindo abstrair uma série de detalhes computacionais.

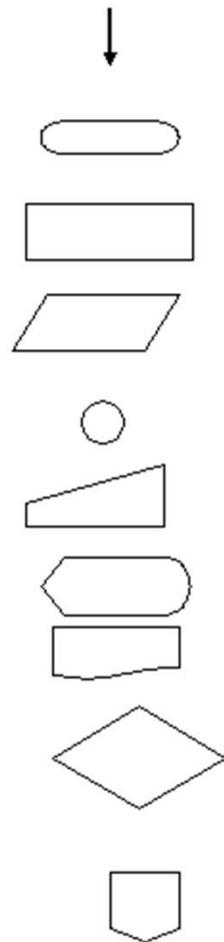
Sequência

- Convenção com o objetivo de reger o fluxo de execução do algoritmo, determinando qual a primeira ação a ser executada e qual ação vem a seguir
 - Linear, de cima para baixo (igual lemos um texto)
 - O fluxo de execução pode ser alterado

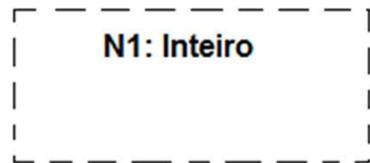
Representação de um Algoritmo

- É uma linha de raciocínio, que pode ser descrita de forma:
 - Gráfica
 - Chapin
 - Fluxograma
 - Textual
 - Pseudocódigo

Fluxograma



- FLUXO DE DADOS**
Indica o sentido do fluxo de dados. Conecta os demais símbolos
- TERMINAL**
Indica o INÍCIO ou FIM de um processamento
Exemplo: Início do algoritmo
- PROCESSAMENTO**
Processamento em geral
Exemplo: Cálculo de dois números
- ENTRADA/SAÍDA (Genérica)**
Operação de entrada e saída de dados
Exemplo: Leitura e Gravação de Arquivos
- DESVIO (conector)**
Permite o desvio para um ponto qualquer do programa
- ENTRADA MANUAL**
Indica entrada de dados via Teclado
Exemplo: Digite a nota da prova 1
- EXIBIR/SAÍDA**
Mostra informações ou resultados
Exemplo: Mostre o resultado do cálculo
- DECISÃO**
Permite elaborar processos de decisão
- CONECTOR DE PAGINA**
Permite informar de qual pagina vem o fluxograma



Declaração de Variáveis

Pseudocódigo

- Forma de representação de algoritmos rica em detalhes e assemelha-se à forma em que os programas são escritos
 - Também conhecida como português estruturado, ou portugol

Algoritmo *<nome_do_algoritmo>*

<declaração_de_variáveis>

<subalgoritmos>

Início

<corpo_do_algoritmo>

Fim

Métodos para Construção de Algoritmos

- A construção de algoritmos é bastante particular, pois envolve o raciocínio de cada indivíduo, suas experiências anteriores, hábitos e criatividade
- Embora não hajam regras rígidas e específicas, há algumas *dicas*

Métodos para Construção de Algoritmos

1. Ler atentamente o enunciado
2. Retirar do enunciado a relação das entradas de dados
3. Retirar do enunciado a relação das saídas de dados
4. Determinar o que deve ser feito para transformar as entradas determinadas nas saídas especificadas
5. Construir o algoritmo/programa
6. Testar o algoritmo/programa

Desenvolvimento Top-Down

- Antes do desenvolvimento é necessário um processo de raciocínio, que vai da análise do problema dado, passando pela elaboração de um esboço, até no fim, a elaboração de um algoritmo detalhado

Sintaxe e semântica

- A sintaxe de uma linguagem é o conjunto de regras que definem quais as relações válidas entre os componentes da linguagem
 - Também conhecida como gramática
- A semântica define o significado de cada frase da linguagem
- **Exemplo de erro sintático:** Café é myito bon.
- **Exemplo de erro semântico:** Café andou até a estrada.

Projeto de programa

- O algoritmo é a estratégia de resolução do problema posta em papel de modo estruturado. Esta estrutura deve ser possível de ser transformada em um programa que seja eficiente para resolver tal problema.

Implementação

- Chamamos de implementação a codificação de um algoritmo em uma linguagem de programação.

Tipos de dados primitivos

- Inteiro, ponto flutuante (real)
- Lógico (Boolean)
- Caracteres (Char)
- literal*

Exercício de fixação

- Usando diagramas, escreva algoritmos para resolver os seguintes problemas:
 1. Ler dois números inteiros e imprimir o produto.
 2. Ler dois números em ponto flutuante e imprima a saída truncada da divisão dos valores (use: `truncar(v1/v2)`).
 3. Ler um número real e imprimir a terça parte desse número.
 4. Ler um número inteiro. Se for maior do que 17, imprima 'maior'.

- 5. Ler um número inteiro. Enquanto for menor do que 1 imprima: “Número inválido”

-