



# **Algoritmos: Conceitos e formas de representação**

**Disciplina: Algoritmos e Programação**  
**Curso: Engenharia de Computação**

Professora: Mariza Miola Dosciatti  
mariza@utfpr.edu.br

# Objetivos

- Compreender o conceito de algoritmo.
- Entender a ideia de algoritmo.
- Conhecer os elementos fundamentais de um algoritmo.
- Saber as formas básicas de organizar as instruções de um algoritmo.
- Conhecer formas de representação de algoritmos.
- Aprender um método para o desenvolvimento de algoritmos.
- Itens da ementa (Plano de Ensino):
  - Conceito de algoritmo e programação.
  - Algoritmos: representação, técnicas e estruturas de elaboração.

# Sumário

## **1. Algoritmo – Conceito**

- 1.1. Algoritmo – Estrutura básica
- 1.2. Algoritmo – Método para a construção
- 1.3. Algoritmo – Considerações
- 1.4. Algoritmo – Elementos fundamentais

## **2. Representação de algoritmos**

- 2.1. Descrição narrativa
- 2.2. Fluxograma
- 2.3. Português estruturado
- 2.4. Linguagem de programação

# 1. Algoritmo: Conceito

- **Procedimentos para vir à UTFPR:**
  - Condição:
    - Morar em uma casa.
  - Passos:
    - Fechar a porta da casa.
    - Fechar o portão.
    - Andar até o ponto de ônibus.
    - Apanhar o ônibus e chegar na universidade.
    - Dirigir-se à sala de aula.

# 1. Algoritmo: Conceito (cont.)

- **Algoritmo (mais detalhado) para vir para a UTFPR:**
  - **Fechar porta da casa.**
    - Outras pessoas permanecem em casa e a porta pode ficar aberta.
    - Outras pessoas permanecem em casa e fecharão a porta.
  - **Fechar o portão.**
    - **Gato quer entrar → reconhecimento do gato.**
      - Se gato conhecido e pode permanecer no jardim e quintal da casa. Então, deixar o gato entrar e fechar o portão.
      - Se gato desconhecido ou não deve permanecer no jardim e quintal da casa. Então, não deixar o gato entrar (ou colocá-lo para fora) e fechar o portão.
  - **Andar até o ponto de ônibus.**
    - **Agendado com colega para apanhar ônibus juntos.**
      - Se o ônibus chegar antes do colega.
        - » Apanhar o ônibus.
        - » Aguardar o colega para apanharem outro ônibus.
        - » Aguardar e ligar para o colega.
        - » Apanhar o ônibus e ligar para o colega.
        - » ....
  - **Ir para a sala de aula.**
    - **Ter verificado (ou saber) a sala correta.**

# 1. Algoritmo: Conceito (cont.)

- Algoritmos fornecem uma descrição detalhada e rigorosa das instruções (ações) para a resolução de problemas.

Definem a sequência de instruções computacionais que representa a solução de problemas. É a essência da programação de computadores.

# 1. Algoritmo: Conceito (cont.)

- **Exemplo:** Algoritmo para ligar de um telefone público – sequência de instruções.

## Início

1. Tirar o fone do gancho;
2. Ouvir o sinal de linha;
3. Inserir o cartão;
4. Teclar o número desejado;
5. Conversar;
6. Desligar;
7. Retirar o cartão;

## Fim



Neste algoritmo todas as instruções são realizadas e na sequência definida.

# 1. Algoritmo: Conceito (cont.)

- Algoritmo para ligar de um telefone público – sequência de instruções.

E se telefone estiver com defeito?

## Início

1. Tirar o fone do gancho;
2. **Se** ouvir o sinal de linha, **então**
  1. Inserir o cartão;
  2. Teclar o número desejado;
  3. Conversar;
  4. Desligar;
  5. Retirar o cartão;
3. **Senão**
  1. Ir para o próximo telefone;

## Fim

Neste algoritmo não são todas as instruções que são realizadas. Quais serão realizadas depende do resultado de um teste lógico (verdadeiro/falso)



# 1. Algoritmo: Conceito (cont.)

- Algoritmo para ligar de um telefone público – repetição  
E se o próximo telefone também estiver com defeito?

## Início

### 1. Repita

1. Tirar o fone do gancho;
2. **Se** ouvir o sinal de linha, **então**
  1. Inserir o cartão;
  2. Teclar o número desejado;
  3. Conversar;
  4. Desligar;
  5. Retirar o cartão;

### 3. Senão

1. Ir para o próximo telefone;

### 2. Até fazer a ligação desejada

## Fim

Neste algoritmo as instruções serão repetidas até ser obtido um determinado resultado (verdadeiro) de um teste lógico realizado.

# 1. Algoritmo: Conceito (cont.)

- **Um algoritmo é uma sequência finita de ações ou instruções** para executar uma tarefa, alcançar um objetivo ou obter uma saída desejada para quaisquer entradas válidas, **visando resolver um problema**.

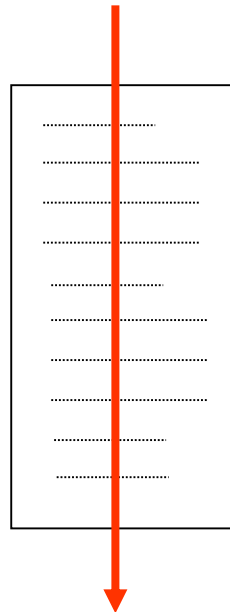


# 1.1. Algoritmo: Estrutura básica

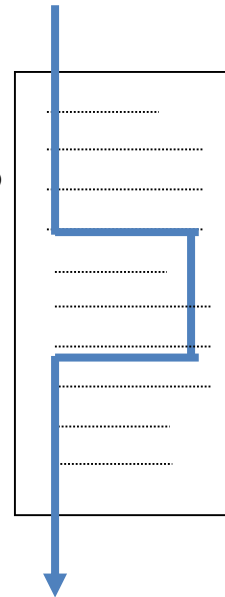
- **Entrada**
  - Dados necessários para realizar as instruções.

- **Processamento**
  - Instruções.

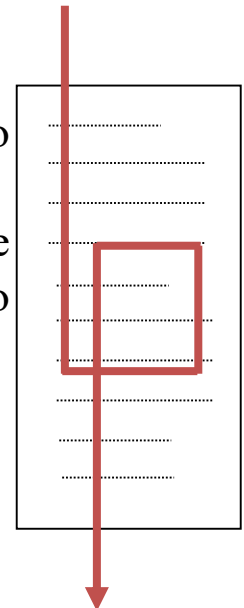
fluxo  
de execução  
sequencial



fluxo  
de execução  
com controle  
de decisão



fluxo  
de execução  
com controle  
de repetição



- **Saída**
  - Resultado: texto e dados (formatados).

## 1.2. Algoritmo: Método para a construção

- Entender o problema;
- Retirar do problema as entradas e as saídas de dados necessárias;
- Determinar o que deve ser feito para transformar as entradas nas saídas desejadas;
  - Determinar o tipo de dado a ser manipulado, definindo as variáveis e as constantes necessárias;
  - Definir as instruções e as estruturas de decisão e de repetição necessárias;
  - Definir as fórmulas e as operações a serem realizadas;
- Apresentar os resultados;
- Verificar se as instruções definidas resolvem o problema da maneira esperada;
  - Realizar testes, como teste de mesa por exemplo, para verificar o resultado.

# 1.3. Algoritmo: Considerações

- No início do algoritmo colocar o enunciado do problema;
- Comentar as partes mais complexas ou relevantes do algoritmo;
- Utilizar espaços e linhas em branco para melhorar a legibilidade do algoritmo, mas sem abusar dos mesmos;
- Escolher nomes representativos para os identificadores;
- Uma instrução por linha é suficiente;
- Utilizar indentação (recuo no início da linha de código).

## 1.4. Algoritmo: Elementos fundamentais

- **De acordo com o paradigma de programação estruturada, os elementos fundamentais são:**
  - a) Tipos de dados;
  - b) Variáveis e constantes;
  - c) Operadores aritméticos, relacionais e lógicos;
  - d) Instruções de entrada e saída;
  - e) Estrutura sequencial;
  - f) Estruturas de controle decisão;
  - g) Estruturas de controle repetição;
  - h) Estruturas de dados homogêneos;
  - i) Funções.

## 2. Representação de Algoritmos

- **Exemplos de formas de representação de um algoritmo:**
  - **Descrição narrativa**
    - Palavras em linguagem natural: português.
  - **Representação gráfica**
    - Fluxograma (diagrama de blocos).
  - **Português estruturado**
    - Pseudocódigo – intermediário entre o português e uma linguagem de programação.

## 2.1. Descrição narrativa

- **Descrição narrativa** ou **linguagem natural** consiste no **uso de frases para expressar ações a serem realizadas**.
- Apresenta a facilidade de a linguagem ser conhecida e o inconveniente da ambiguidade de termos.
- Procedimento para elaborar um algoritmo em descrição narrativa:
  - **Definição dos dados de entrada**
  - **Processamento (instruções a serem realizadas)**
  - **Definição dos dados de saída**



## 2.1. Descrição narrativa (cont.)

- **EXEMPLO:** Receita de bolo
  - Misture os ingredientes.
  - Unte a forma com manteiga.
  - Despeje a mistura na forma.
  - Leve ao forno.
  - Enquanto não dourar, deixe o bolo no forno.
  - Retire do forno.
  - Deixe esfriar.

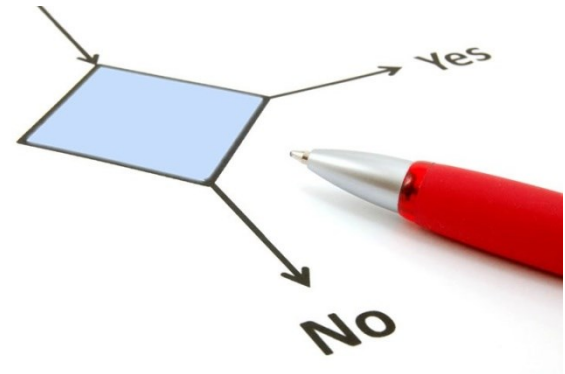


## 2.1. Descrição narrativa (cont.)

- **EXEMPLO:** **Calcular a soma de dois números**
  - Apresentar na tela uma mensagem para o usuário informando-o que deverá digitar dois números do tipo inteiro.
  - Após ler a mensagem, o usuário deverá inserir os números por meio do teclado, que serão lidos e armazenados em duas variáveis diferentes.
  - Em seguida, será aplicada a operação de soma nos dois operandos informados pelo usuário e o resultado será armazenado em uma terceira variável.
  - O resultado será apresentado na tela para o usuário e o algoritmo será finalizado.


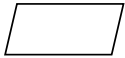
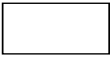
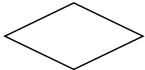


## 2.2. Fluxograma

- Um **fluxograma**, **diagrama de fluxo** ou **diagrama de blocos**, é uma representação gráfica que **utiliza formas geométricas** ligadas por setas **para indicar as instruções** que devem ser seguidas para resolver um problema.
- A principal **função de um fluxograma** é facilitar a visualização da sequência das instruções de um algoritmo.



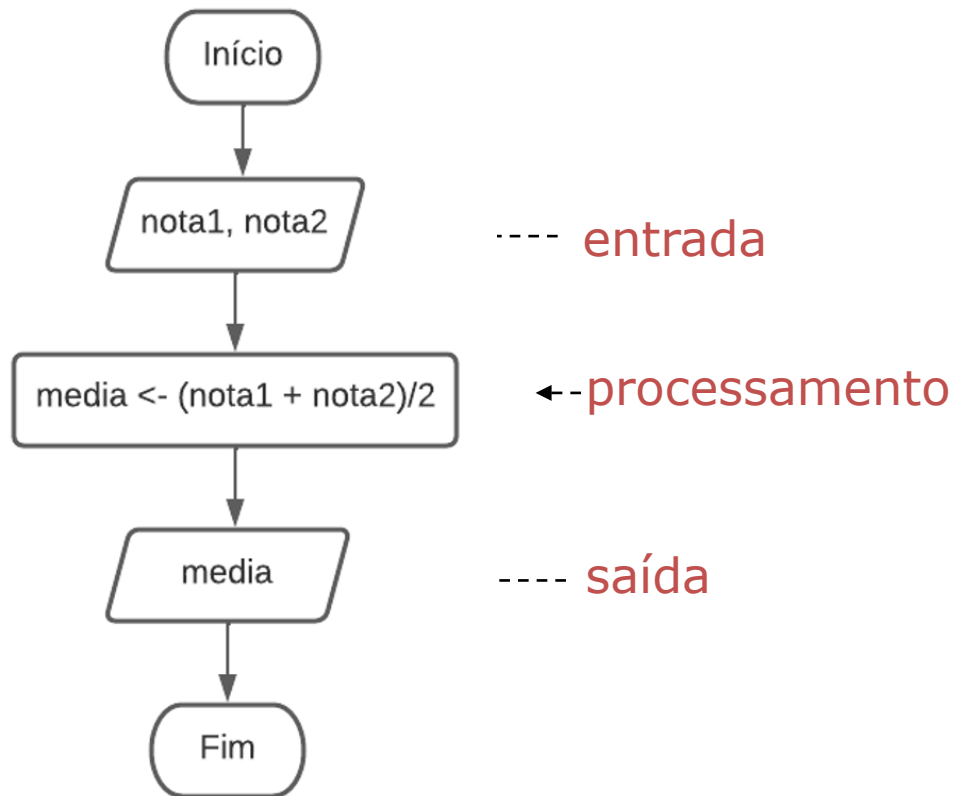
## 2.2. Fluxograma (cont.)

- Exemplos de símbolos (norma ISO 5807):**

Símbolo	Descrição
	<b>Terminal:</b> início e fim do algoritmo.
	<b>Entrada e saída:</b> Receber e mostrar informações. Ler dados para armazenar em variáveis e mostrar dados contidos em variáveis. Escrever texto.
	<b>Processamento:</b> realizar operações com variáveis e constantes; executar as instruções contidas em estruturas de decisão e de repetição.
	Representar as condições de uma estrutura de decisão.
	Conector para agrupar fluxos.
	<b>Linha:</b> conector direcionado entre símbolos.

## 2.2. Fluxograma (cont.)

- Fluxograma Estrutura Sequencial.
- Problema: **Calcular a média de duas notas.**



Teste de mesa:

Nota1	Nota2	Media
10	9	9,5
7	8	7,5
10	10	10

## 2.3. Português estruturado

- **Português Estruturado** (Pseudocódigo, Pseudolinguagem, Portugol) é uma forma de representar algoritmos utilizando um conjunto de termos da língua portuguesa para expressar instruções.

## 2.3. Português estruturado (cont.)

- Forma geral da representação de um algoritmo em português estruturado:

**Algoritmo** <nome\_do\_algoritmo>

<declaração\_de\_variáveis>

**Início**

<instruções>

**Fim**

- A forma de representação de algoritmos em português estruturado assemelha-se às linguagens de programação.

## 2.3. Português estruturado (cont.)

- **EXEMPLO:**

**Algoritmo** "Media"

// Calcula a média aritmética de duas notas

**Var**

// Declarações das variáveis  
nota1, nota2, media: real

**Inicio**

// Entrada de dados  
escreva ("Informe a primeira nota: ")  
leia (nota1)  
escreva ("Informe a segunda nota: ")  
leia (nota2)

//Processamento de dados  
media <- (nota1 + nota2 ) / 2

//Saída de dados  
escreva ("A média aritmética é: ", media)

**Fimalgoritmo**



# Aplicação das formas de representação

As formas de representação de algoritmos são utilizadas em vários contextos na área da Computação, mas principalmente em enunciados e resoluções de problemas em provas e livros da área.

# Referências

- **FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. Lógica de programação: A construção de algoritmos e estruturas de dados.** São Paulo: Makron, 1993.

# Dúvidas

- ???