

# Algoritmos

## O que é um Algoritmo?

Um **algoritmo** é uma sequência finita de passos bem definidos usados para resolver um problema ou executar uma tarefa específica. É como uma receita que guia a execução de uma atividade, seja ela matemática, computacional ou prática.

## Definição Formal

1. **Conjunto de Regras Finito:** Um algoritmo consiste em um conjunto de instruções que devem ser seguidas para se chegar a uma solução.
2. **Processo Finito:** Um algoritmo deve sempre terminar após um número finito de passos.
3. **Independência de Linguagem:** Algoritmos são independentes de linguagens de programação específicas; são expressos em forma genérica e podem ser implementados em qualquer linguagem.

## Exemplo do Dia a Dia

Imagine cozinhar uma receita. O algoritmo é a lista de passos que você segue:

- Pegar os ingredientes (entrada).
- Misturar os ingredientes seguindo a ordem correta (processamento).
- Finalizar com o prato pronto (saída).

---

## Por que Precisamos de Algoritmos?

Os algoritmos são essenciais em diversas áreas:

1. **Ciência da Computação:** Utilizados para resolver problemas como ordenação, busca de dados e tarefas mais complexas, como inteligência artificial.
2. **Matemática:** Resolvem problemas como cálculos de sistemas lineares ou gráficos.
3. **Engenharia:** Otimizam processos e ajudam no design de sistemas eficientes.
4. **Operações do Dia a Dia:** Desde navegação por GPS até análise de dados financeiros.

Exemplo prático: A busca de um produto em um site de e-commerce utiliza um algoritmo para encontrar os itens relevantes rapidamente.

---

## Características de um Algoritmo

Para que uma sequência de passos seja considerada um algoritmo, ela deve atender aos seguintes critérios:

1. **Clareza e Desambiguidade:** Cada passo deve ser claro e ter um único significado.
  2. **Entradas Bem Definidas:** Um algoritmo pode ou não requerer entradas, mas, se necessário, elas devem ser claras.
  3. **Saídas Bem Definidas:** O algoritmo deve gerar pelo menos um resultado.
  4. **Finitude:** O processo deve terminar após um número limitado de etapas.
  5. **Eficiência e Simplicidade:** Deve ser prático e executável com os recursos disponíveis.
  6. **Independência de Linguagem:** Pode ser descrito em pseudocódigo, fluxogramas ou qualquer representação genérica.
- 

### Exemplo Prático: Soma de Três Números

**Problema:** Criar um algoritmo para somar três números e exibir o resultado.

**Etapas para Resolver o Problema:**

1. Definir as entradas: três números inteiros.
2. Realizar a soma dos números.
3. Exibir o resultado.

**Pseudocódigo do Algoritmo:**

INÍCIO

Declare três variáveis: num1, num2, num3.

Solicite ao usuário os três números e armazene nas variáveis correspondentes.

Calcule a soma: soma = num1 + num2 + num3.

Exiba a soma.

FIM

## 2. Pseudocódigo

### Definição

O pseudocódigo é uma forma de descrever algoritmos utilizando uma linguagem simplificada e intermediária entre a linguagem natural (como português) e uma linguagem de programação. Ele ajuda a estruturar ideias antes de implementar o código em uma linguagem específica.

### Características

- Não segue a sintaxe rígida de uma linguagem de programação.
- Fácil de entender, mesmo para iniciantes.
- Foco na lógica e no fluxo de execução.

## Exemplo de pseudocódigo:

**Problema:** Verificar se um número é par ou ímpar.

Início

    Leia número

    Se número MOD 2 = 0 então

        Escreva "Número é par"

    Senão

        Escreva "Número é ímpar"

Fim

---

## 3. Fluxograma

### Definição

Um fluxograma é uma representação gráfica de um algoritmo ou processo. Ele usa símbolos padronizados para representar diferentes etapas e setas para indicar o fluxo de execução.

### Símbolos Comuns

- **Elipse:** Início ou fim do processo.
- **Retângulo:** Ações ou instruções.
- **Losango:** Decisões ou condições.
- **Setas:** Direção do fluxo.

### Vantagens

- Visualmente intuitivo.
- Útil para identificar problemas no fluxo.
- Auxilia na comunicação entre equipes.

### Exemplo de Fluxograma

**Problema:** Verificar se um número é positivo, negativo ou zero.

### Descrição do Algoritmo:

1. Leia um número.
2. Verifique se o número é maior que zero:
  - Se for, exiba "Número positivo".
3. Caso contrário, verifique se o número é menor que zero:
  - Se for, exiba "Número negativo".
4. Se não for nenhuma das opções acima, exiba "Número é zero".

### Fluxograma:

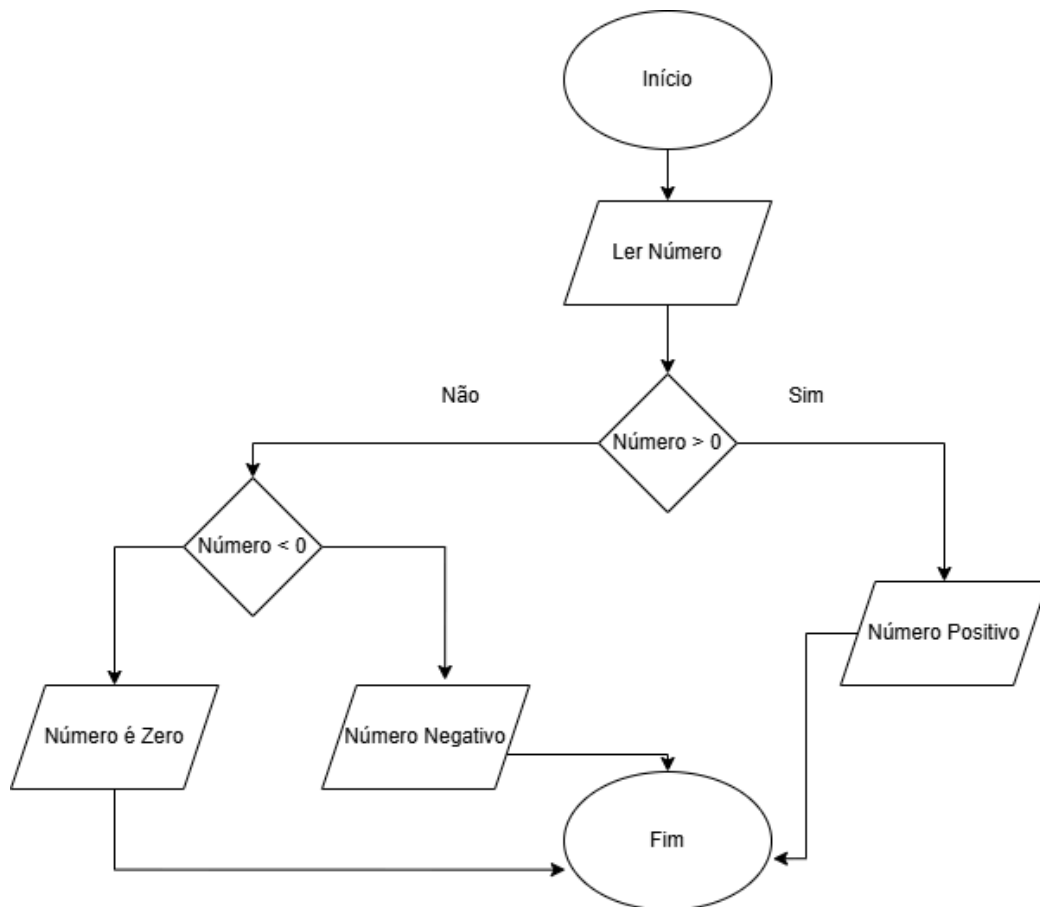


Imagem 1: Exemplo de Fluxograma

---

## Resumo

Um algoritmo é uma sequência de passos lógicos e bem definidos utilizados para resolver um problema ou realizar uma tarefa. Ele está presente tanto em situações do dia a dia

quanto no desenvolvimento de soluções computacionais, sendo fundamental para a programação.

O pseudocódigo surge como uma maneira simplificada de descrever algoritmos, utilizando uma linguagem intermediária que não exige a rigidez de uma sintaxe de programação. Sua função é facilitar a organização das ideias e o planejamento da lógica antes da implementação em uma linguagem específica. Por exemplo, para determinar se um número é par ou ímpar, podemos usar um pseudocódigo que lê o número, verifica a condição de divisibilidade por 2 e apresenta o resultado.

Complementando o pseudocódigo, o fluxograma oferece uma representação visual do fluxo lógico de um algoritmo. Por meio de símbolos padronizados, como retângulos para ações e losangos para decisões, o fluxograma permite visualizar o processo de resolução de um problema de maneira clara e intuitiva. Um exemplo prático seria um fluxograma que verifica se um número é positivo, negativo ou zero, utilizando condições para determinar a saída apropriada.

A compreensão de algoritmos, pseudocódigo e fluxogramas é essencial para estruturar o pensamento lógico e desenvolver soluções eficientes. A prática desses conceitos reforça a capacidade de criar programas robustos e bem planejados, promovendo o aprendizado contínuo na programação.

### **1. Khan Academy - Algoritmos e Fluxogramas**

KHAN ACADEMY. *Algoritmos e Fluxogramas*. Disponível em: <<https://www.khanacademy.org>>. Acesso em: 15 nov. 2024.

### **2. GeeksforGeeks - Introduction to Algorithms**

GEEKSFORGEEKS. *Introduction to Algorithms*. Disponível em: <<https://www.geeksforgeeks.org/introduction-to-algorithms/>>. Acesso em: 15 nov. 2024.

### **3. Livro: Algoritmos: Teoria e Prática**

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. *Algoritmos: Teoria e Prática*. 3. ed. Rio de Janeiro: Elsevier, 2022.