

# Algoritmos

Projeto de Certificadora de Competência Comum

André Agostinis  
Gabriel Victor Tavares  
Leila Minello  
Luan Almeida

---

André Luis dos  
Santos Domingues

# Conteúdo

- **Definição de algoritmo**

- Utilizações
- Características
- Exemplos práticos

- **Pseudocódigo**

- Definição e características
- Exemplo

- **Fluxograma**

- Definição e características
- Exemplo

- **Referências**

# Definição

## 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.

## Definição Formal

- **Conjunto de regras finito:** consiste em um conjunto de instruções que devem ser seguidas para se chegar a uma solução.
- **Processo finito:** deve sempre terminar após um número finito de passos.
- **Independência de linguagem:** são independentes de linguagens de programação específicas; podem ser implementados em qualquer linguagem.



O algoritmo é a lista de passos que você segue, como numa receita:

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

## Os algoritmos são essenciais em diversas áreas:

- **Ciência da Computação:** utilizados para resolver problemas como ordenação, busca de dados e até inteligência artificial.
- **Matemática:** resolvem problemas como cálculos de sistemas lineares ou gráficos.
- **Engenharia:** otimizam processos e ajudam no design de sistemas eficientes.
- **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.

# Utilizações

Por que precisamos  
de algoritmos?

# Características de um algoritmo

**Critérios para que uma sequência de passos seja considerada um algoritmo:**

- **Clareza e desambiguidade:** passos devem ser claros e ter um único significado.
- **Entradas bem definidas:** se um algoritmo precisar de uma entrada, ela deve ser clara.
- **Saídas bem definidas:** o algoritmo deve gerar pelo menos um resultado.
- **Finitude:** o processo deve terminar após um número limitado de etapas.
- **Eficiência e simplicidade:** deve ser prático e executável com os recursos disponíveis.
- **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:

- Definir as entradas: três números inteiros.
- Realizar a soma dos números.
- 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**

# 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

# 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.

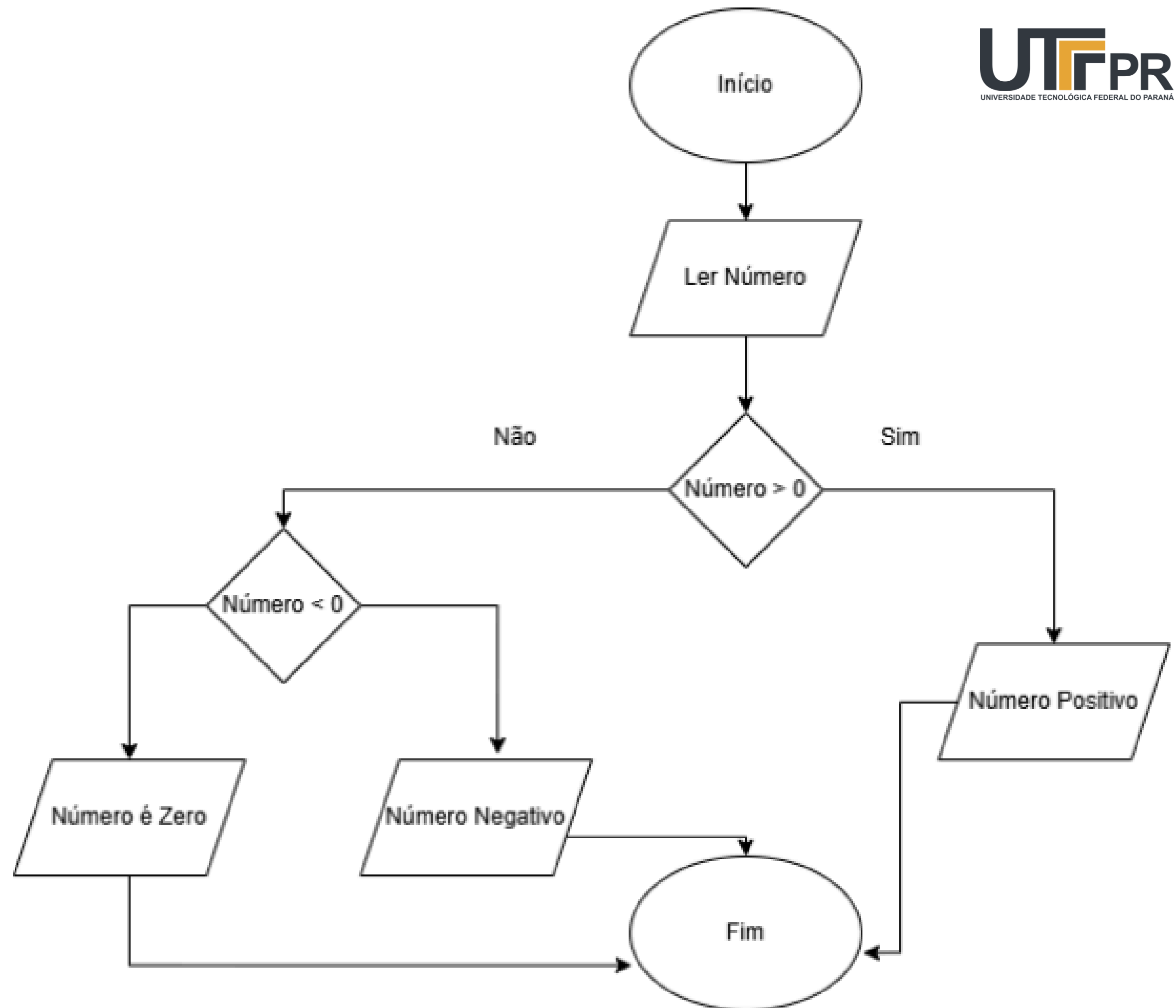


# Exemplo prático

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

Descrição:

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



# Referências

## 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.