



## Lógica de Programação

Bem-vindo a este guia sobre lógica de programação, a base para criar soluções computacionais!



## O que é lógica de programação?

### Linguagem computacional

É uma maneira de "falar" com computadores, instruindo-os a executar tarefas específicas.

### Sequência de passos

Envolve definir uma sequência lógica de instruções para que o computador possa processar informações e gerar resultados.

## Componentes básicos da lógica de programação

### Variáveis

Contêineres para armazenar dados, como números, textos ou valores booleanos.

### Operadores

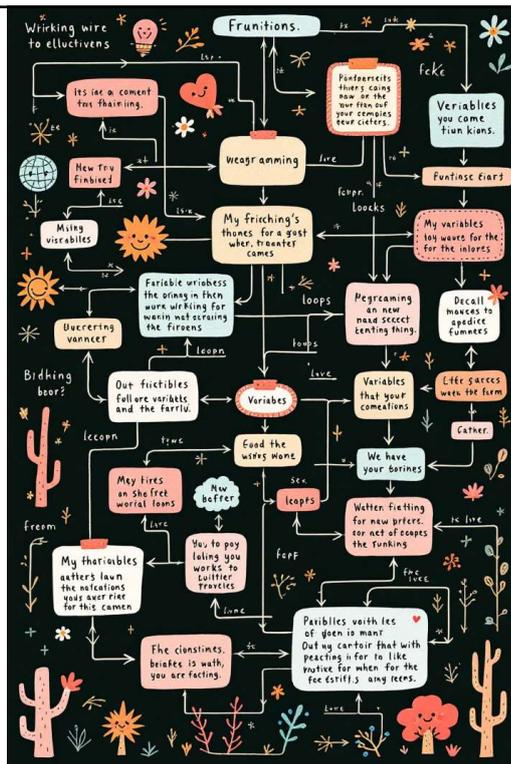
Símbolos que realizam operações matemáticas, de comparação ou lógicas sobre dados.

### Estruturas de controle

Controlam o fluxo de execução do programa, determinando quais instruções serão executadas.

### Funções

Blocos de código reutilizáveis que executam tarefas específicas, tornando o código mais organizado.



## Estruturas de controle (seleção e repetição)

1

### Seleção

Decisões que alteram o fluxo do programa com base em condições, como "se" ou "senão".

2

### Repetição

Executam blocos de código várias vezes até que uma condição seja satisfeita, como "enquanto" ou "para".



# Algoritmos e pseudocódigo

1

## Algoritmo

Conjunto de instruções passo a passo para resolver um problema, independente da linguagem de programação.

2

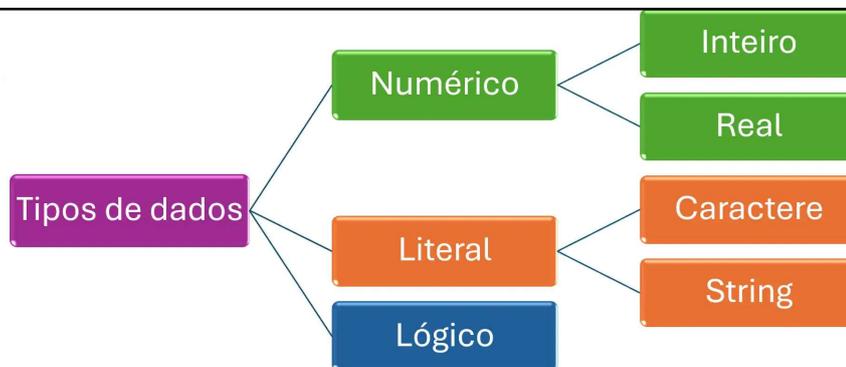
## Pseudocódigo

Descrição textual de um algoritmo, utilizando uma linguagem informal para facilitar a compreensão.

3

## Exemplo

Para calcular a média de duas notas, você precisa somá-las e dividir por 2.



# Variáveis e tipos de dados

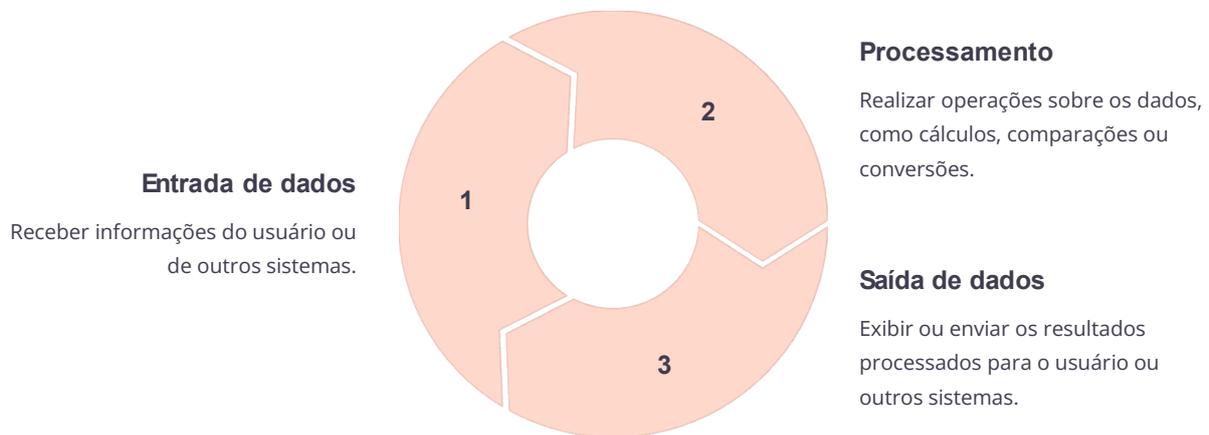
## Variáveis

Nomes que referenciam locais na memória do computador para armazenar dados.

## Tipos de dados

Definem o tipo de informação que uma variável pode armazenar, como números inteiros, textos ou valores booleanos.

## Manipulação de dados e operações



## Boas práticas em lógica de programação

1

### Legibilidade

Escreva código claro e fácil de entender para você e outros programadores.

2

### Modularidade

Divida o código em funções menores para facilitar a organização e reutilização.

3

### Comentários

Explique o propósito do seu código com comentários para melhorar a compreensão.

4

### Testes

Verifique se o código funciona como esperado com testes para garantir a qualidade.