

# Etapa 1 - Introdução a Algoritmos e Pseudocódigo

Trilha: Lógica de Programação

# O que é um Algoritmo (Algorithm)?

- Um **algoritmo** é uma sequência finita de **passos lógicos** que resolvem um problema ou executam uma tarefa.
- Está presente em **todas as áreas da computação**.
- É **independente de linguagem de programação**.
- Base da **lógica de programação** e do **pensamento computacional (computational thinking)**.

## Exemplo do mundo real:

- Fazer café:
  1. **Ferver água**
  2. **Colocar o pó no filtro**
  3. **Despejar água quente**
  4. **Servir o café**

Isso é um algoritmo cotidiano!

# Características de um Algoritmo

- **Finitude (Finiteness)** → Deve terminar após um número limitado de passos.
- **Definição (Definiteness)** → Cada instrução deve ser clara e não ambígua.
- **Entrada (Input)** → Dados que o algoritmo recebe.
- **Saída (Output)** → Resultados gerados.
- **Efetividade (Effectiveness)** → Cada passo deve ser possível de executar.

# Representação de Algoritmos

Existem várias formas de representar um algoritmo:

- **Linguagem natural** (texto comum)
- **Fluxograma (Flowchart)**
- **Pseudocódigo (Pseudocode)** — a mais usada no ensino
- **Código fonte (Source Code)** em uma linguagem de programação real

# O que é Pseudocódigo (Pseudocode)?

- É uma **forma estruturada e simples** de descrever algoritmos.
- Não segue a sintaxe de uma linguagem real, mas **parece código**.
- Foca na **lógica**, não na execução.
- Usa palavras-chave intuitivas como:
  - inicio, fim, se, entao, senao, enquanto, para, leia, escreva.

## **Exemplo:**

inicio

  leia A, B

  soma  $\leftarrow$  A + B

  escreva "Resultado:", soma

fim

# Vantagens do Pseudocódigo

- Fácil de entender mesmo sem saber programar.
- Ideal para **planejar algoritmos** antes de codificar.
- **Facilita o aprendizado** de estruturas de controle e lógica.
- Serve como **ponte** entre a ideia e a implementação.

# Estrutura Básica de um Algoritmo

- Um pseudocódigo genérico segue esta estrutura:

```
algoritmo "nome_do_algoritmo"
```

```
inicio
```

```
    // Declaração de variáveis
```

```
    // Instruções e comandos
```

```
fim
```

- **Exemplo:**

```
algoritmo "calcular_media"
```

```
inicio
```

```
    leia nota1, nota2
```

```
     $media \leftarrow (nota1 + nota2) / 2$ 
```

```
    escreva "Média =", media
```

```
fim
```

# **Etapas para Construção de um Algoritmo**

- 1. Compreender o problema**
- 2. Definir entradas e saídas**
- 3. Dividir em passos lógicos**
- 4. Testar mentalmente**
- 5. Representar em pseudocódigo ou fluxograma**
- 6. Validar com exemplos reais**



# Boas Práticas

- Nomeie variáveis de forma **significativa**.
- Use **indentação** para facilitar leitura.
- Comente (//) trechos importantes.
- Mantenha os passos **curtos e claros**.

## **Exemplo com comentários:**

inicio

// Ler dois números e somar

leia x, y

$z \leftarrow x + y$

escreva "Soma =", z

fim

# Conclusão

- Todo programa começa com um **algoritmo** bem estruturado.
- O **pseudocódigo** ajuda a **pensar logicamente** antes de programar.
- Dominar essa base é essencial para seguir na trilha de lógica de programação.