

# CAPÍTULO 1 – INTRODUÇÃO À LÓGICA E À PROGRAMAÇÃO

**Carga horária:** 4h (2h teoria + 2h prática)

**Etapa:** Fundamentos do Pensamento Computacional

**Objetivo:** Introduzir o conceito de **lógica** como base do raciocínio humano e da programação, explorando como o pensamento estruturado se aplica na resolução de problemas do cotidiano. O aluno compreenderá a importância da **ordem, sequência e coerência das ações** antes da escrita de qualquer código, utilizando situações reais e representações gráficas.

## 1.1. O que é Lógica?

A **lógica** é a ciência que estuda as formas corretas de pensar e raciocinar. Ela permite que organizemos as ideias de forma estruturada, coerente e objetiva, para chegar a conclusões verdadeiras.

- ✓ Segundo **Irving M. Copi (1978)**, filósofo e professor da Universidade de Michigan:

*“A lógica é o estudo dos métodos e princípios usados para distinguir o raciocínio correto do incorreto.”*



*“Lógica: ciência que estuda os princípios e métodos do raciocínio correto.”*

Em resumo, usar lógica é **organizar o pensamento** de forma sequencial e coerente, garantindo que cada ação leve a um resultado previsível.

Sem lógica, não há programação, pois o computador precisa de instruções claras e exatas.

## 1.2. A Lógica no Cotidiano

A lógica não está restrita à computação — ela é usada constantemente em nossa vida diária.

Sempre que seguimos uma **ordem correta de passos**, aplicamos lógica.

**Exemplo (cotidiano):**

**Situação:** Abrir uma porta trancada

1. “Pegar a chave;”
2. “Inserir na fechadura;”
3. “Girar a chave;”
4. “Abrir a porta.”

Se invertermos a ordem (tentar abrir antes de inserir a chave), o resultado será incorreto.

- ✓ Essa **sequência de ações** é um exemplo prático de **raciocínio lógico**.



1. Colocar a chave na fechadura.
2. Girar a chave para destrancar a porta.
3. Girar a maçaneta.
4. Puxar a porta para abri-la.

“Lógica no cotidiano: sequência de passos para resolver um problema.”

Outros exemplos comuns:

- Fazer café: a água deve ferver antes de coar.
- Dirigir: engatar a marcha antes de acelerar.
- Tomar banho: ligar o chuveiro antes de entrar.

Essas ações mostram que o **pensamento lógico** está presente em tudo o que fazemos, e que a programação apenas transforma esse raciocínio em linguagem de máquina.

### 1.3. Lógica de Programação

A **Lógica de Programação** é o processo mental que transforma ideias em instruções organizadas.

Antes de escrever código, é preciso **entender o problema e planejar a sequência de passos** que levará à solução.

*“A lógica de programação é o conjunto de regras e técnicas que orientam o raciocínio humano na criação de soluções computacionais.”*



“Pensamento lógico → Algoritmo → Programa → Aplicação.”

✓ **Resumindo:**

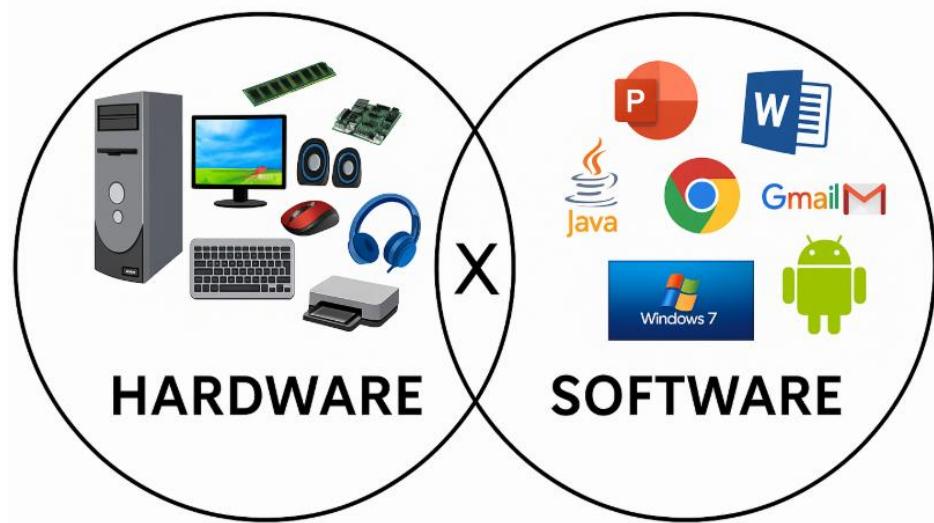
A lógica é **independente de linguagem**.

Quem domina lógica aprende qualquer linguagem (Visualg, C, JavaScript, Python), com facilidade.

#### 1.4. Hardware e Software

Todo sistema computacional é formado por duas partes principais:

Componente	Descrição	Exemplo
<b>Hardware</b>	Parte física — o que se pode tocar.	Monitor, teclado, mouse, CPU.
<b>Software</b>	Parte lógica — conjunto de instruções que orienta o hardware.	Windows, Linux, Visualg, Arduino IDE.



O **hardware** é o corpo do computador.

O **software** é a mente que o controla.

Um depende do outro: o software **comanda** e o hardware **executa**.

## 1.5. Tipos de Software

Os softwares são classificados de acordo com sua função principal.

Tipo de Software	Função Principal	Exemplo
Sistema	Controla o funcionamento do computador.	Windows, Linux, Android.
Aplicativo	Realiza tarefas específicas.	Word, Excel, Paint.
Programação	Permite criar novos programas.	Visual Studio Code, Arduino IDE.



Esses três tipos se complementam:

O **software de programação** cria **aplicativos**, que são executados sobre o **software de sistema**.

### 1.6. Algoritmos no Dia a Dia

Antes de escrever qualquer código, o programador precisa **pensar como um solucionador de problemas**.

Isso significa **observar, planejar e definir a sequência correta** para atingir o resultado.

✓ **Exemplo:**

Como calcular a idade de uma pessoa:

1. Solicitar o ano de nascimento;
2. Obter o ano atual;
3. Subtrair o ano de nascimento do ano atual;
4. Exibir o resultado.

Essas quatro etapas formam o **algoritmo** do cálculo da idade.

Na programação, esse tipo de raciocínio é traduzido para comandos que o computador pode compreender.

### 1.7. Fluxograma: o desenho da lógica

O **fluxograma** é uma representação **visual** de um algoritmo. Cada símbolo representa uma etapa, ação ou decisão dentro de um processo.

Símbolo	Significado	Uso
	Início/Fim	Marca onde o algoritmo começa e termina
	Processo	Representa uma ação ou cálculo
	Decisão	Indica uma condição (sim/não)
	Fluxo	Mostra a direção do processo

O fluxograma ajuda o aluno a **visualizar a lógica antes de codificar**. No próximo capítulo, ele aprenderá a criá-los no papel e, em seguida, no Visualg.

### 1.8. Atividade Guiada – O Pensamento Lógico em Ação

O raciocínio lógico é a base de toda programação. Para compreender sua importância, observe o exemplo a seguir.

✓ **Situação: preparar um copo de suco natural**

Veja abaixo uma sequência desordenada de ações:

1. Ligar o liquidificador
2. Colocar açúcar
3. Descascar a fruta
4. Colocar água
5. Servir o suco

Parece simples, mas há um problema: **a ordem está incorreta**.

O resultado dessa sequência seria um suco mal preparado ou até impossível de fazer, já que o liquidificador seria ligado antes mesmo de colocar os ingredientes.

✓ **Sequência lógica correta**

- 1 Descascar a fruta
- 2 Colocar a fruta no liquidificador
- 3 Adicionar a água
- 4 Acrescentar o açúcar
- 5 Ligar o liquidificador
- 6 Servir o suco

**Perceba:** Cada passo depende do anterior. Se a ordem for alterada, o processo falha.

Essa sequência é, na prática, um **algoritmo do cotidiano**.

✓ **Representando em Fluxograma**



**Pensamento de preparo do suco natural**

✓ **Descrição do fluxograma:**

Um fluxograma simples, mostrando:

**Elipse (Início)** → “Começar”

**Retângulo (Processo)** → “Descascar a fruta”

**Retângulo (Processo)** → “Colocar no liquidificador”

**Retângulo (Processo)** → “Adicionar água e açúcar”

**Retângulo (Processo)** → “Ligar o liquidificador”

**Retângulo (Processo)** → “Servir o suco”

**Elipse (Fim)** → “Fim do processo”

- **Setas** → conectam cada etapa na sequência lógica correta.

**Conclusão:**

- O fluxograma ajuda a visualizar o raciocínio do processo.
- Cada símbolo mostra o que está sendo feito, e a ordem das setas garante que o processo aconteça corretamente.
- Na programação, o computador executa exatamente essas etapas, nem mais, nem menos.

### 1.9. Exploração Guiada – Criando Sequências Lógicas

Agora é a sua vez de praticar a construção de raciocínios lógicos.

**Objetivo:** Aprender a identificar a ordem correta das ações e aplicar o raciocínio lógico em situações do cotidiano.

**Atividade:**

1. Escolha uma tarefa comum (ex.: escovar os dentes, ligar o computador, fazer um sanduíche).
2. Escreva os passos **fora de ordem**, como no exemplo anterior.
3. Depois, reorganize de forma **lógica e coerente**, numerando corretamente.
4. Desenhe um **fluxograma simples** com as etapas corretas.

**Dica:** Use setas e símbolos conforme o modelo estudado (elipse para início/fim, retângulo para processo, losango para decisão).

👉 Onde você usa lógica sem perceber?

---

---

---

---

👉 O que aconteceria se o computador seguisse suas instruções exatamente como foram ditas?

---

---

---

---

👉 Por que planejar é mais importante do que começar a programar?

---

---

---

---

## 1.10. Atividade de Treino – Pensando com Lógica

Após observar e praticar, é hora de treinar sozinho.

**Objetivo:** Consolidar o raciocínio sequencial, descrevendo processos de forma clara e ordenada.

**Atividade:**

1. Escolha uma tarefa cotidiana diferente das anteriores.
2. Escreva o passo a passo completo.
3. Verifique se há lógica entre as etapas.
4. Reescreva, se necessário.
5. Crie um fluxograma representando o processo.

**Reflexão:** Se outra pessoa seguir suas instruções exatamente, ela conseguirá o mesmo resultado?

Esse é o princípio da **programação lógica**.

**Dica:** use verbos de ação (pegar, ligar, verificar, abrir) e mantenha a sequência coerente.

## 1.11. Conclusão

“Um bom programador não nasce sabendo desenhar fluxogramas. Ele aprende primeiro a **pensar com lógica**.” — Hilton Elias Tomaszeswisc dos Santos (*Instrutor e Autor, 2025*)

**Ao final deste capítulo, você deverá ser capaz de:**

- Organizar ideias em sequências lógicas;

- Compreender a relação entre hardware e software;
- Entender o conceito de algoritmo;
- Preparar-se para representar processos em fluxogramas no próximo capítulo.