



Lógica de Programação e Algoritmos

Prof. William C. Augostonelli (Billy)
03/02/2025

M1.

Introdução à Lógica de Programação

- ✓ Conceito de lógica e algoritmos
- Passos para resolver problemas computacionais
 - Representação de algoritmos (Fluxogramas e Pseudocódigo)
 - Introdução a ambientes de programação (Portugol, Python, C, etc.)

A2.

Passos para resolver problemas computacionais

-
- Compreensão do problema
 - Planejamento da solução
 - Desenvolvimento do algoritmo
 - Implementação e testes
 - Ajustes e otimizações

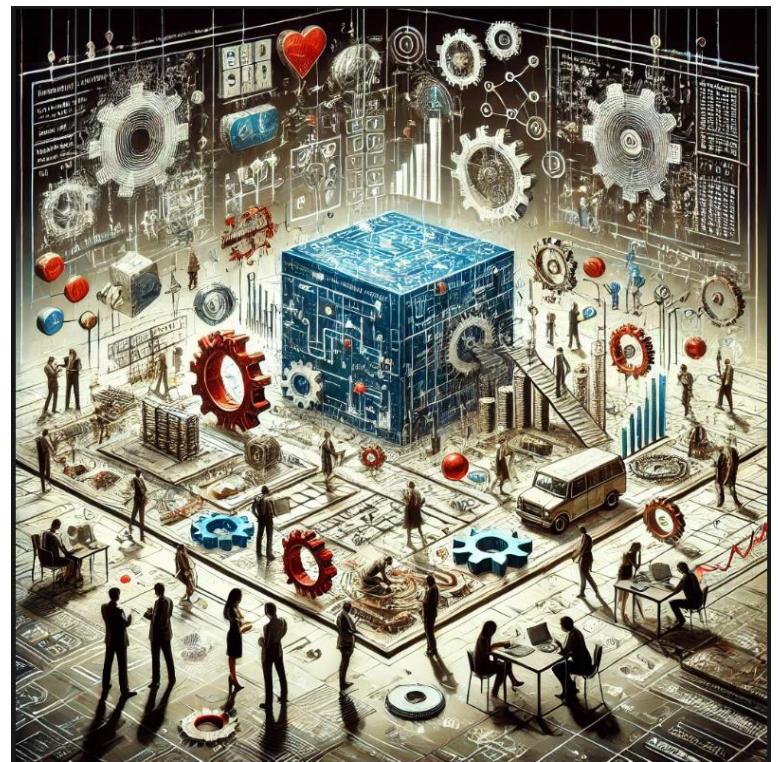
• Objetivo

- Entender os passos necessários para resolver problemas computacionais
- Aplicar cada um dos passos para criar soluções eficientes e corretas

Passos para resolver problemas computacionais

Compreensão do Problema

- O primeiro passo ao resolver um problema é **compreendê-lo bem**.
- Sem entender corretamente o que precisa ser feito, a solução provavelmente será incorreta
- **Leia** o enunciado do problema com atenção, destaque os pontos chave e pergunte
 - O que é dado?
 - O que é necessário encontrar?
 - Quais são as condições e restrições?



Passos para resolver problemas computacionais

- Exemplo
 - **Problema:** você tem R\$ 100,00 e quer comprar o máximo de itens possíveis em uma loja, onde cada item custa R\$ 15,00.
Quantos itens você pode comprar?
 - **Dado:** você tem R\$ 100,00 e cada item custa R\$ 15,00
 - **O que é necessário?:** encontrar quantos itens podem ser comprados
 - **Condicional:** não pode gastar mais do que R\$ 100,00

Planejamento da solução

- Após compreender o problema, o próximo passo é **planejar** como resolver a situação
- O planejamento envolve criar uma estratégia ou um plano de ação
- **Pense** nas operações matemáticas, lógicas ou etapas necessárias para atingir o objetivo
- **Exemplo**

Plano para o problema

1. Divida o valor total (R\$ 100,00) pelo custo de um item (R\$ 15,00)
2. A parte inteira do resultado será o número de itens que você pode comprar

Desenvolvimento do algoritmo

- Depois de planejar a solução, o próximo passo é criar o **algoritmo**, que é a sequência de instruções para resolver o problema
- O algoritmo pode ser escrito de várias formas
 - Pseudocódigo
 - Fluxograma
 - Em forma de código (Linguagem de programação)

Desenvolvimento do algoritmo

- Exemplo em algoritmo – pseudocódigo

Iniciar

Definir o total de dinheiro como 100

Definir o preço do item como 15

Dividir o total do dinheiro pelo preço do item

Exibir o resultado (número de itens)

Fim



Implementação e Testes

- **IMPLEMENTAÇÃO** é o processo de transformar o algoritmo em código de programação
- Após implementar, é crucial realizar **TESTES** para verificar se a solução está correta
 - **Teste** com diferentes entradas e verifique se o programa gera resultados esperados



Implementação e Testes

- Exemplo (em Python)



A screenshot of a Jupyter Notebook interface. The title bar shows "Untitled0.ipynb" and "Salvando...". The menu bar includes "Arquivo", "Editar", "Ver", "Inserir", "Ambiente de execução", "Ferramentas", and "Ajuda". On the left, there are icons for code (+ Código), text (+ Texto), search (Q), and cell types (x, {x}, key). The main area contains the following Python code:

```
total_dinheiro = 100
preco_item = 15
itens_comprados = total_dinheiro // preco_item
print("Você pode comprar", itens_comprados, "itens.")
```

The output cell shows the result: "Você pode comprar 6 itens."

- **Testes**

- Teste com o valor de R\$ 75,00 e preço de R\$ 5,00
- Teste com valores diferentes de dinheiro e preços de itens
- Teste com entrada diferente de valores de dinheiro

Ajustes e Otimizações

- Nos testes pode ser necessário fazer **ajustes** e após os testes, você deve verificar se é possível **otimizar**
 - **Ajustes:** Corrija bugs ou falhas encontradas nos testes
 - **Otimização:** melhorar a performance ou tornar o código mais simples, sem perder a clareza
 - **Exemplo de ajuste**
 - Se o valor de R\$ 100,00 não é exatamente divisível por R\$ 15,00 o algoritmo pode ser ajustado para arredondar o número de itens de forma mais inteligente, ou para informar que sobrou um valor que não é suficiente para comprar outro item

Exercícios

- **Exercício 1: Compreensão do problema**

- Leia o enunciado a seguir e compreenda o que é necessário fazer.
Problema: João tem R\$ 50,00 e quer comprar o maior número possível de ingressos para um show. O preço de cada ingresso é de R\$ 12,00. Quantos ingressos ele pode comprar e quanto sobrou de dinheiro?

- **Exercício 2: Planejamento da Solução**

- Planeje a solução para o seguinte problema.
Problema: um aluno tem uma nota final composta por 3 provas: 5, 7 e 8. Calcule a média final e verifique se o aluno passou (média maior ou igual a 6). Crie um plano para resolver isso

- **Exercício 3: Desenvolvimento de Algoritmo**

- Escreva um algoritmo para verificar se um número dado é múltiplo de 3 e 5.

- **Exercício 4: Ajustes e Otimizações**

- Após testar sua solução do exercício 3, tente otimizar o código. Existe uma maneira mais eficiente de verificar se o número é múltiplo de 3 e 5? Se sim, implemente a melhoria.

