

Módulo Algoritmos e Linguagens de Programação

Professor: Marcos Brandão

Versão: 2026

APRESENTAÇÃO

A programação está presente em praticamente todas as tecnologias utilizadas atualmente: aplicativos de celular, sistemas bancários, sites, jogos, carros, equipamentos industriais e dispositivos inteligentes. Antes de aprender uma linguagem de programação, é fundamental desenvolver o raciocínio lógico e compreender como solucionar problemas de maneira organizada.

Os algoritmos representam a base da programação. Eles descrevem uma sequência lógica de passos para resolver um problema, independentemente da linguagem utilizada. Ao dominar esse conceito, o estudante estará preparado para aprender linguagens como C, Python, Java, JavaScript, PHP e muitas outras.

Esta apostila apresenta os fundamentos da lógica de programação, os elementos básicos dos algoritmos, estruturas de controle, vetores, modularização e os princípios da programação estruturada, preparando o estudante para desenvolver programas simples e organizados.

UNIDADE I

LÓGICA DE PROGRAMAÇÃO E CONSTRUÇÃO DE ALGORITMOS

(Páginas 1 a 5)

1. O que é Lógica de Programação?

Lógica de programação é a capacidade de organizar pensamentos e criar uma sequência de instruções para solucionar um problema.

Programar não significa apenas escrever códigos, mas analisar uma situação, identificar uma solução e transformá-la em instruções que possam ser executadas pelo computador.

Exemplo do cotidiano

Problema: Fazer um café.

Algoritmo:

1. Colocar água na chaleira.
2. Aquecer a água.
3. Colocar o pó de café no filtro.
4. Despejar a água quente.
5. Servir o café.

Essa sequência organizada representa um algoritmo.

2. O que é um Algoritmo?

Algoritmo é uma sequência finita e ordenada de passos para resolver um problema ou executar uma tarefa.

Todo algoritmo deve possuir:

- início;
- processamento;
- resultado;
- fim.

Características de um bom algoritmo

- Clareza
- Organização
- Objetividade
- Eficiência

- Correção

3. Formas de Representar Algoritmos

Existem diversas formas de representar algoritmos.

Linguagem Natural

Descrição utilizando o idioma português.

Exemplo:

"Solicite o nome do usuário e exiba uma mensagem de boas-vindas."

Fluxograma

Representação gráfica utilizando símbolos.

Principais símbolos:

Símbolo	Função
Oval	Início/Fim
Retângulo	Processo
Paralelogramo	Entrada/Saída
Losango	Decisão

Pseudocódigo (Portugol)

É a forma mais utilizada no ensino de programação.

Exemplo:

Algoritmo "BoasVindas"

Inicio

Escreva("Digite seu nome:")
Leia(nome)

Escreva("Olá ", nome)

Fim

4. Elementos Básicos

Todo programa utiliza informações chamadas dados.

Os principais tipos são:

Tipo	Exemplo
Inteiro	10
Real	7.5
Caracter e	"A"
Cadeia	"Bahia"
Lógico	Verdadeiro/Falso

Variáveis

Variáveis armazenam informações na memória.

Exemplo:

idade ← 20

nome ← "Carlos"

media ← 8.5

Constantes

São valores que não mudam durante a execução.

Exemplo:

PI = 3.14159

Operadores

Aritméticos

+

-

*

/

%

Relacionais

>

<

>=

<=

==

<>

Lógicos

E

OU

NÃO

5. Entrada e Saída de Dados (E/S)

Todo programa recebe informações e produz resultados.

Entrada

Leia(nome)

Saída

Escreva(nome)

Exemplo completo:

Leia(idade)

Escreva("Sua idade é ", idade)

Atividades da Unidade I

1. Defina algoritmo.
2. Explique a diferença entre algoritmo e programa.
3. Cite três formas de representar algoritmos.
4. O que é uma variável?
5. Escreva um algoritmo que leia o nome do aluno e mostre uma mensagem de boas-vindas.

UNIDADE II

PROGRAMAÇÃO ESTRUTURADA

(Páginas 6 a 10)

1. Paradigma Estruturado

A programação estruturada organiza o código em blocos bem definidos, facilitando a leitura, manutenção e reutilização.

Ela utiliza três estruturas básicas.

Estrutura Sequencial

As instruções são executadas em ordem.

Exemplo:

Leia(a)

Leia(b)

soma $\leftarrow$ a + b

Escreva(soma)

Estrutura Condicional

Permite tomar decisões.

Exemplo:

Se nota $\geq$ 6 Então

Escreva("Aprovado")

Senão

Escreva("Reprovado")

FimSe

Estrutura de Repetição

Executa comandos diversas vezes.

Enquanto

Enquanto contador ≤ 10 Faça

Escreva(contador)

contador $\leftarrow$ contador + 1

FimEnquanto

Para

Para i de 1 até 10 Faça

Escreva(i)

FimPara

2. Arrays (Vetores)

Arrays armazenam vários valores do mesmo tipo utilizando apenas um nome.

Exemplo:

notas[5]

Representação:

Índice	Valor
0	7
1	8
2	10
3	5
4	9

Exemplo:

Para i de 0 até 4

Leia(notas[i])

FimPara

3. Matrizes

São vetores com duas dimensões.

Exemplo:

notas[3][4]

Representam tabelas com linhas e colunas.

Muito utilizadas para:

- planilhas;
 - jogos;
 - imagens;
 - bancos de dados.
-

4. Modularização

Programas grandes devem ser divididos em módulos menores.

Cada módulo realiza uma tarefa específica.

Vantagens:

- organização;
 - manutenção;
 - reutilização;
 - facilidade de testes.
-

Procedimentos

Executam tarefas sem retornar valores.

Exemplo:

Procedimento Mensagem()

Escreva("Bem-vindo")

FimProcedimento

Funções

Retornam um resultado.

Exemplo:

Função Soma(a,b)

Retorne a+b

5. Linguagens de Programação Estruturadas

Os algoritmos podem ser implementados em diversas linguagens.

Exemplos:

- C
- Pascal
- Python
- Java
- PHP
- JavaScript

Apesar das diferenças na sintaxe, todas utilizam os mesmos conceitos estudados nesta apostila.

Boas Práticas de Programação

- ✓ Utilizar nomes significativos para variáveis.
 - ✓ Escrever código organizado.
 - ✓ Criar comentários.
 - ✓ Evitar repetição de código.
 - ✓ Testar os programas.
 - ✓ Corrigir erros antes da entrega.
-

Resumo Geral

Ao concluir esta apostila, o estudante deverá compreender que:

- algoritmos são a base da programação;
- lógica é mais importante que decorar comandos;
- programas são compostos por entrada, processamento e saída;
- estruturas de decisão e repetição controlam o fluxo do programa;
- vetores armazenam vários dados;
- modularização facilita o desenvolvimento;
- diferentes linguagens utilizam os mesmos conceitos fundamentais.

Esses conhecimentos servirão de base para disciplinas posteriores de desenvolvimento de sistemas, programação web, banco de dados e desenvolvimento de aplicações.

Exercícios Finais

1. Explique o conceito de algoritmo.
2. Diferencie variável e constante.
3. Escreva um algoritmo que leia duas notas e calcule a média.
4. Escreva um algoritmo que informe se um número é par ou ímpar.
5. Faça um algoritmo utilizando uma estrutura de repetição para exibir os números de 1 a 20.
6. Crie um vetor para armazenar as notas de cinco alunos e calcule a média da turma.
7. Explique a diferença entre procedimento e função.
8. Pesquise uma linguagem de programação estruturada (como C, Python ou Pascal) e descreva:
 - o suas principais características;
 - o áreas de aplicação;
 - o vantagens e limitações.

Glossário

Termo	Significado
Algoritmo	Sequência lógica de passos para resolver um problema.
Lógica de Programação	Forma organizada de pensar e solucionar problemas computacionais.
Pseudocódigo	Representação textual de um algoritmo, independente de linguagem.
Variável	Espaço de memória destinado ao armazenamento de dados que podem variar durante a execução do programa.
Constante	Valor que permanece inalterado durante a execução do programa.
Estrutura Condicional	Bloco que permite ao programa tomar decisões com base em uma condição.
Estrutura de Repetição	Bloco utilizado para repetir comandos enquanto uma condição for satisfeita ou por um número definido de vezes.
Array (Vetor)	Estrutura de dados que armazena diversos elementos do mesmo tipo utilizando um único nome e índices.
Matriz	Estrutura bidimensional formada por linhas e colunas.
Modularização	Técnica de dividir um programa em funções e procedimentos menores para facilitar o desenvolvimento e a manutenção.
Programação Estruturada	Paradigma de programação baseado nas estruturas sequencial, condicional e de repetição, promovendo código organizado, legível e reutilizável.