

01 - Apresentação da Disciplina

Pedro Cacique

Graduado em Engenharia Elétrica pela
Universidade Federal de Uberlândia (UFU)

Mestre em Engenharia Elétrica e
Computação pela UFU

Doutor em Engenharia Elétrica e
Computação pelo Mackenzie

 Distinguished Educator

 Professional Learning Specialist

Teoria

Quinta-feira | 07:30 às 09:10 | Prédio 33, sala 101

Prática

Turma 01G11 - Terça-feira | 09:20 às 11:00 | Prédio 31, sala 401 (Prof. Alexandre)

Turma 01G12 - Sexta-feira | 09:20 às 11:00 | Prédio 31, sala 402 (Profa. Elisângela)

Plano de Ensino

Ementa

Estudo e desenvolvimento de algoritmos envolvendo comandos de atribuição, condicionais e de repetição, tendo com ênfase a resolução de problemas em ordem crescente de complexidade.

Implementação de algoritmos utilizando linguagem de programação imperativa.

Objetivos

Fatos e Conceitos

- Conhecer o conceito de algoritmo computacional
- Identificar passos para soluções de problemas elementares e formalizá-los através de algoritmos
- Avaliar e comparar soluções algorítmicas para problemas elementares
- Conhecer estruturas de programação de uma linguagem imperativa e aplicá-los na implementação de algoritmos

Objetivos

Procedimentos e Habilidades

- Construir algoritmos computacionais para problemas elementares
- Implementar algoritmos em uma linguagem de programação imperativa
- Configurar e utilizar ambientes de implementação de algoritmos
- Simular implementações de algoritmos para avaliação de funcionamento e detecção de erros

Objetivos

Atitudes, Normas e Valores

- Reconhecer a importância dos algoritmos para resolução de problemas.
- Reconhecer a importância da linguagem de programação na implementação de algoritmos.
- Reconhecer a importância do teste de algoritmos.
- Reconhecer a área de programação como um suporte essencial na construção de sistemas computacionais.
- Perceber e superar dificuldades inerentes ao pensamento algorítmico.

Conteúdo Programático

UNIDADE I: FUNDAMENTOS DE ALGORITMOS E PROGRAMAÇÃO

1.1. Problemas e soluções

1.2. Algoritmos e exemplos de notação (fluxogramas, pseudocódigo)

1.3. Programas e algoritmos

Conteúdo Programático

UNIDADE II: VARIÁVEIS, TIPOS DE DADOS, EXPRESSÕES, ATRIBUIÇÃO E ESTRUTURA SEQUENCIAL

2.1. Constantes e variáveis

2.2. Tipos (numéricos, booleanos, caractere)

2.3. Operadores e expressões matemáticas

2.4. Operador de atribuição

2.5. Estrutura sequencial

2.6. Problemas envolvendo variáveis, tipos de dados, expressões, atribuição e estrutura sequencial

Conteúdo Programático

UNIDADE III: ESTRUTURAS DE DECISÃO

3.1. Operadores relacionais e lógicos e suas tabelas

3.2. Estrutura de Seleção Simples

3.3. Estrutura de Seleção Composta

3.4. Encadeamento de estruturas de decisão

3.5. Problemas envolvendo estruturas de decisão

Conteúdo Programático

UNIDADE IV: ESTRUTURAS DE REPETIÇÃO

- 4.1. Estrutura de repetição com teste no início
- 4.2. Estrutura de repetição com teste no final
- 4.3. Estrutura de repetição com variável de controle
- 4.4. Encadeamento de estruturas de repetição
- 4.5. Problemas com estruturas de repetição

Conteúdo Programático

UNIDADE V: FUNÇÕES

5.1. Definição e Uso de Funções

5.2. Aplicações em problemas

Conteúdo Programático

UNIDADE VI: REPRESENTAÇÃO DE LISTAS

6.1. Notação de Listas

6.2. Operações sobre Listas

6.3. Exemplos e Aplicações

METODOLOGIA

- Aulas expositivas
- Aulas práticas em laboratórios
- Suporte extraclasses através de plantões de professores e monitores
- Utilização do ambiente Moodle
- Listas de Exercícios e Projetos Práticos

AVALIAÇÃO

Nota 1 (N1) composta de:

- Prova Parcial – T1 (60%)
- Atividades de Laboratório (Lab1) (10%)
- Projeto de Laboratório (Proj1) (20%)
- Atividades dos módulos 1 e 2 (Curso: Python Essentials 1 – CISCO) (PEC1) (10%)

AVALIAÇÃO

Nota 2 (N2) composta de:

- Prova Parcial – T1 (50%)
- Atividades de Laboratório (Lab1) (15%)
- Projeto de Laboratório (Proj1) (25%)
- Atividades dos módulos 1 e 2 (Curso: Python Essentials 1 – CISCO) (PEC1) (10%)

AVALIAÇÃO

$$MI = (N1 + N2)/2 + NP$$

- **Nota de participação (NP):** até 1,0 (um), sendo 0,5 referente a nota da prova integrada (AvaliA) e 0,5 referente ao progresso no curso Python Essentials 2 da parceria com a Cisco Network Academy.

AVALIAÇÃO

CRITÉRIO 1:

se $MI \geq 6,0$ e $FREQUENCIA \geq 75\%$, 🍑

CRITÉRIO 2:

se $FREQUENCIA \geq 75\%$ e $(MI + PROVA\ FINAL) / 2 \geq 6.0$, 🍑

BIBLIOGRAFIA

BÁSICA

DIERBACH, C. Introduction to Computer Science Using Python: A Computational ProblemSolving Focus.1.ed. New York: Wiley, 2012.

MENEZES, N.N.C. Introdução à Programação com Python: algoritmos e lógica de programação para iniciantes. São Paulo: Novatec, 2a.ed, 2014.

ZELLE, J.M. Python Programming: An Introduction to Computer Science, 2nd Edition, Franklin, Beedle & Associates Inc, 2009.

BIBLIOGRAFIA

COMPLEMENTAR

FORBELLONE, A. L. V.; EBERSPACHER, H. F. **Lógica de Programação: A Construção de Algoritmos e Estrutura de Dados.** 3. ed. Sao Paulo: Pearson Prentice Hall, 2012.

KINSLEY, H.; MCGUGAN, W. **Introdução ao Desenvolvimento de Jogos em Python com PyGame.** São Paulo: Novatec, 2015.

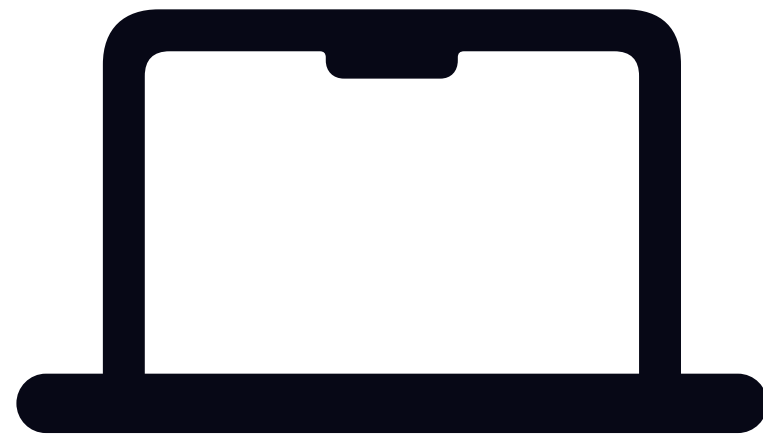
LOPES, A.; GARCIA, G. **Introdução a Programação: 500 Algoritmos.** Rio de Janeiro: Editora Campus, 2002.

PAYNE, B. **Ensine seus filhos a programar.** São Paulo: Novatec, 1a. ed. 2015.

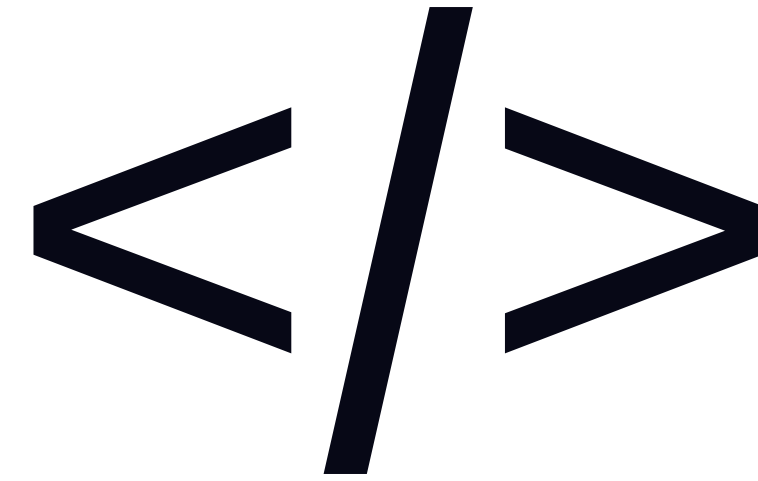
PIVA Jr., D.; NAKAMITI, G.S., ENGELBRECHT, A.M. **Algoritmos e Programação de Computadores.** Rio de Janeiro: Editora Elsevier Ltda, 2012.

Introdução

O computador



Hardware



Software

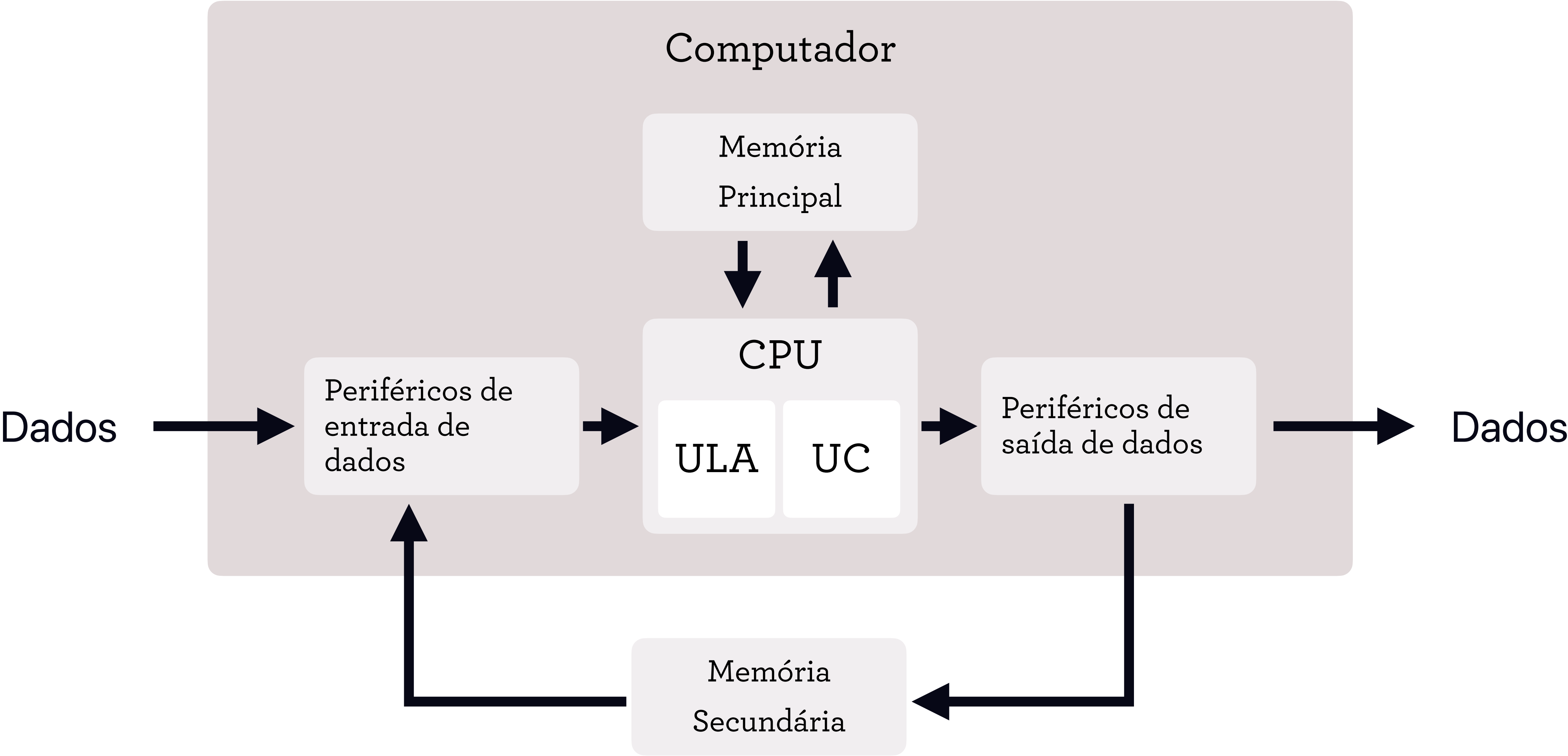
Dados



Processamento



Dados



Organização da Ciência

“A lógica tradicional, desenvolvida por Aristóteles, foi criada originalmente para ajudar pessoas a pensar de forma mais efetiva, através do uso de silogismo, o qual é a base da Matemática e da Computação.”

– (BRACKMANN, 2017)

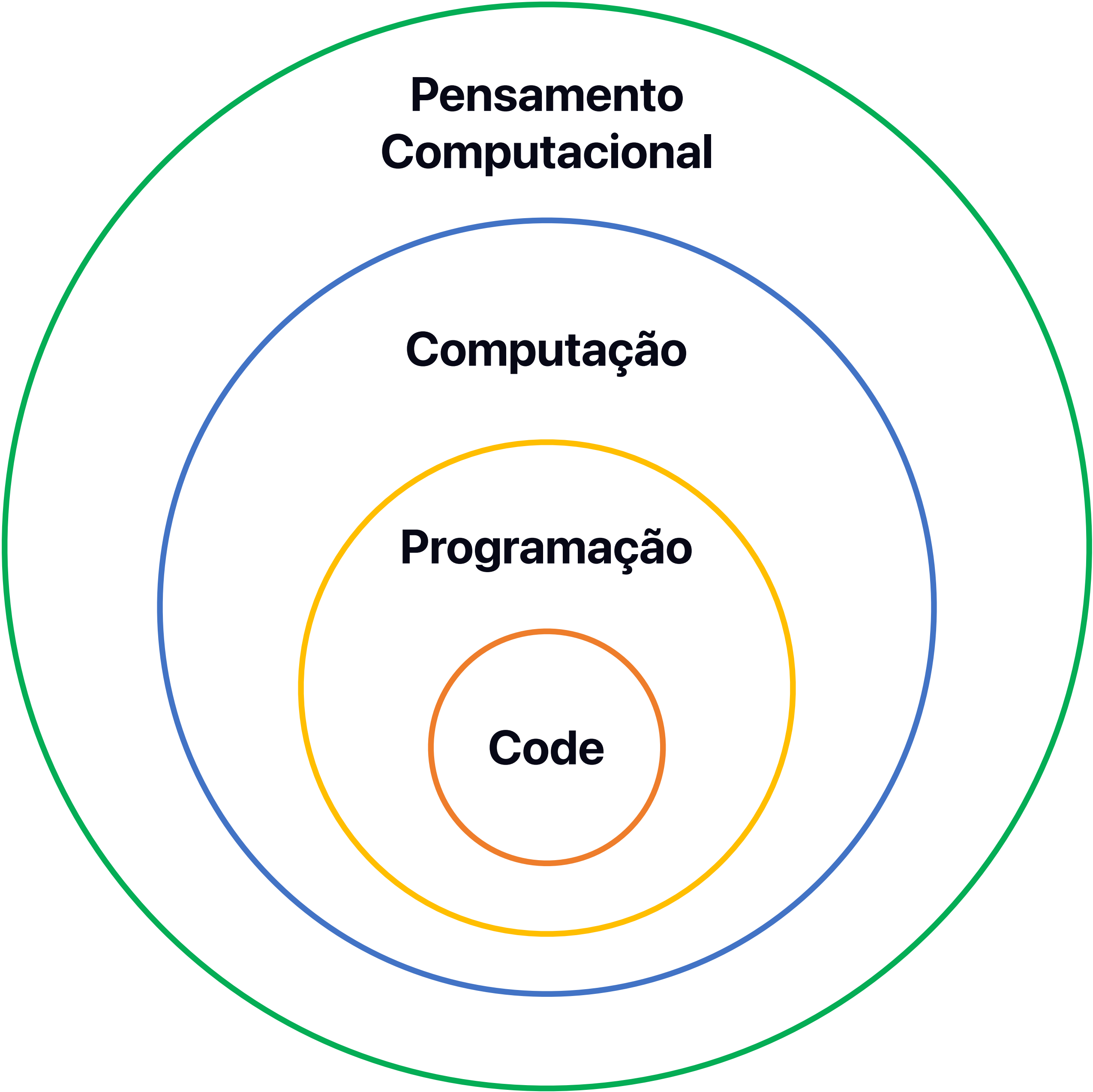


Teoria

Experimentação

Computação

Pensamento Computacional



Decomposição

Reconhecimento de Padrões

Abstração

Algoritmo

Algoritmo

Computacionalmente definimos como um conjunto de regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema em um número finito de passos.

5 propriedades fundamentais para algoritmos

- FINITUDE: deve sempre parar após um número finito de etapas.
- DEFINIÇÃO: devem ser definidos com precisão, as ações especificadas rigorosamente e sem ambiguidades.
- ENTRADA: valores fornecidos ao algoritmo antes que ele inicie.
- SAÍDA: resultados a partir de uma determinada entrada.
- EFICÁCIA: todas as operações devem ser suficientemente básicas para poderem, em princípio, ser feitas com precisão e em um período de tempo finito por um homem usando papel e lápis.

