

AED2 - Algoritmos e Estr. de Dados II

Aula 1: Apresentação e Visão Geral da Disciplina

Prof. Aléssio Miranda Júnior
alessio@cefetmg.br

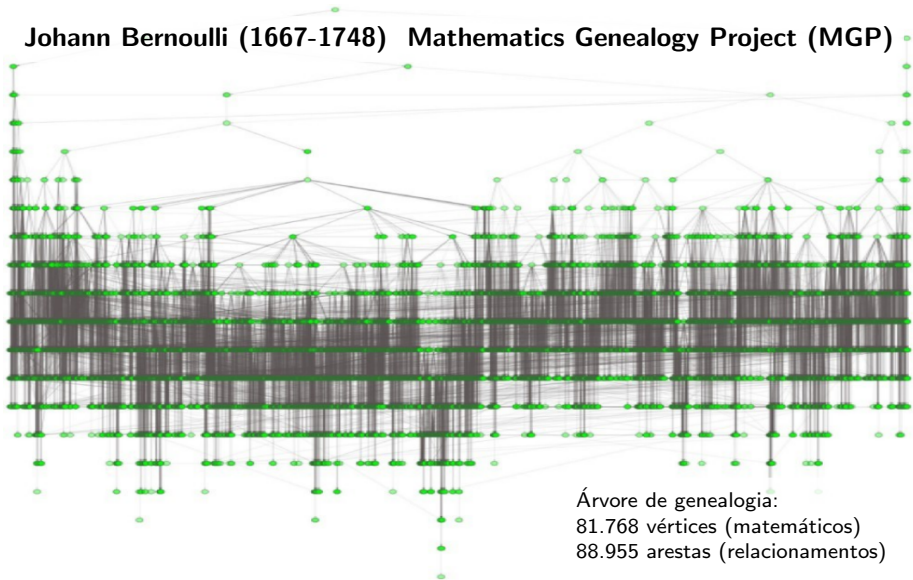
CEFET-MG - Campus Timóteo
Dep. Engenharia de Computação

4 de Março de 2026

- 1 Introdução
- 2 Legado de AED I
- 3 AED II
- 4 Livros
- 5 Estrutura
- 6 Avaliação

- **Professor:** Aléssio Miranda Júnior (DCCC-TIM)
- **Formação:**
 - Ciência da Computação, Doutorado em 2026 (UFMG)
 - Mestre em Ciência da Computação (UFPR)
- **Endereço:** www.alessiojr.com
- **Áreas de pesquisa e interesses:**
 - Empreendedorismo e Startups
 - Competições de Programação
 - Plataformas e Políticas de Software Livre
 - Gamificação em estratégias com I.A

Johann Bernoulli (1667-1748) Mathematics Genealogy Project (MGP)





Fuga/Migração de pessoas

Aplicação de estruturas de dados eficientes para a análise de dados.

1. O Legado de AED I: O Mundo Linear

Em **AED I**, o foco foi **armazenamento sequencial** e fundamentos essenciais:

- **Estruturas Básicas:** Vetores, Listas, Pilhas e Filas.
- **Fundamentos:** Recursividade e noções matemáticas de Complexidade de Algoritmos.
- **Busca e Ordenação:** Sequencial/Binária e algoritmos (Inserção, Bolha, Merge, Quick, etc.).
- **Mapeamento:** Tabelas e funções Hash.

O Limite: Estruturas lineares falham em modelar o mundo real complexo (hierarquias, redes, relacionamentos arbitrários).

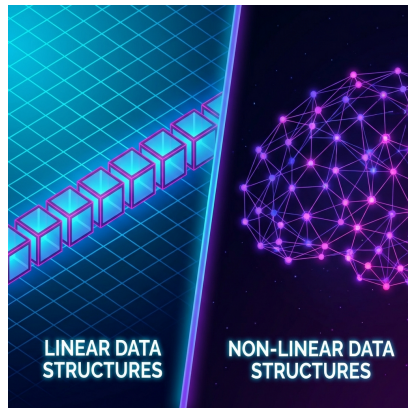


Figure: Linear (AED I) vs Conexo (AED II)

2. O Foco de AED II: Hierarquias, Pesquisa e Grafos

Avançando na Complexidade:

- **Ordenação em Tempo Linear:** Counting, Radix e Bucket Sort.
- **Memória Primária:** Hashing, Union-Find e **Árvores Balanceadas** (AVL, Rubro-Negras).
- **Memória Secundária:** Gerenciamento e pesquisa massiva com **Árvores B** e arquivos indexados.
- **Redes Complexas:** Introdução, modelagem e heurísticas em **Grafos**.



Figure: De lineares para estruturados

Identificação: AED2-2018-2QTPI: 4 - (4 Teóricas)

Dedicação

É muito importante cerca de ~ 4 horas de estudo fora da aula.

Postura e Recomendações:

- → Fall in love with mathematics
- → Be self-motivated
- → Never back down
- → Become a master
- → Be a bookworm
- (*pratique matemática*)
- (*trabalhe com pares*)
- (*seja persistente*)
- (*ensine aos colegas*)
- (*seja leitor ávido*)

- Atualmente existem várias linguagens consideradas para este tipo de disciplina: **Python, C, C++, Java, Haskell, Ruby**.
- Também vários **paradigmas de programação** podem ser considerados (e.g., procedural, orientado a objetos, etc.).
- É importante saber e entender a linguagem **C/C++**.
- Entretanto, em tempos atuais, **Java** está mais em evidência no mercado.
- Tradicionalmente é utilizada a linguagem **C/C++**.
- Nesta disciplina **usaremos JAVA**.

Características da linguagem C:

- Não possui suporte nativo à orientação a objetos.
- Linguagem de nível intermediário: controle mais direto do hardware, porém também suporta estruturas complexas.
- Gerenciamento de memória **explícito** (uso de ponteiros e `malloc/free`).
- Detecção de erro explícita (sem blocos `try/catch`).
- Maior performance direcional do programa final.
- Maior dificuldade de manutenção à medida que os sistemas crescem.

Sobre a linguagem de programação JAVA

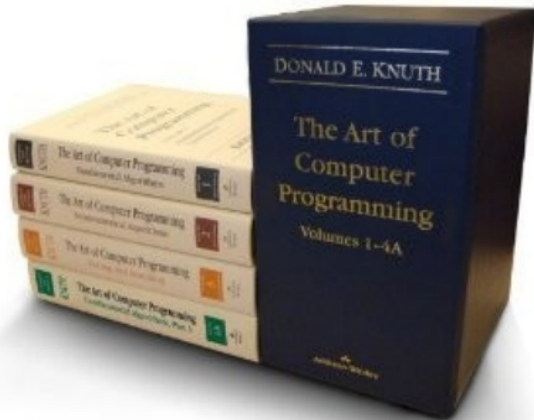
- Precisa instalar um compilador e uma máquina virtual (*JVM*) no seu computador.
- Será usada a **JDK 11**, mas não é obrigatório o seu uso estrito (versões mais recentes são compatíveis).
- A utilização de uma **IDE** é **desnecessária** (mas recomendada para projetos maiores).
- Durante a aula usarei **IntelliJ** ou **VSCode**.

Configuração do Ambiente

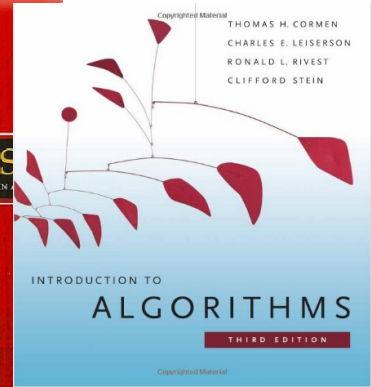
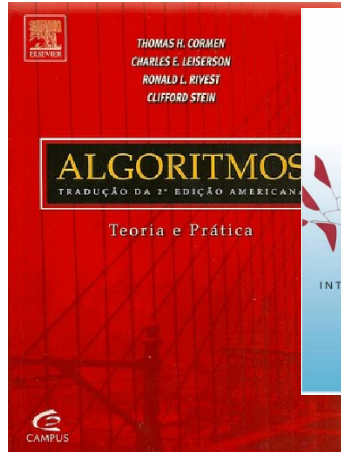
- **No Linux:** é padrão/fácil instalação via repositório.
- **No Mac/OS:** terá que instalar as ferramentas de desenvolvimento.
- **No Windows:** baixar e instalar o executável da **JDK**.

- KNUTH, Donald E. *The Art of Computer Programming*. Addison-Wesley.

“Gates afirmou que qualquer pessoa que lesse e compreendesse a série de livros, considerada a ‘bíblia da programação’, deveria enviar-lhe um currículo para trabalhar na Microsoft. Ele destacou a imensa dificuldade e disciplina necessárias para ler a obra, comparando-a a um rito de passagem.”



- CORMEN, Thomas H.
Algoritmos: teoria e prática.
2002.



- GOODRICH, Michael T.; TAMASSIA, Roberto. *Estruturas de dados e algoritmos em JAVA*.



- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L. e STEIN, C. *Introduction to Algorithms*, 3ª edição, MIT Press, 2009.
- ZIVIANI, N. *Projeto de Algoritmos: com implementações em Pascal e C*, 2ª edição, Cengage Learning, 2009.
- FOLK, M.; ZOELLICK B. *File Structures*, 2ª edição, Addison-Wesley, 1992.
- FOLK, M.; ZOELLICK, B.; RICCARDI, G. *File Structures, An Object-Oriented Approach Using C++*, 3ª edição, Addison-Wesley, 1998.
- SEDGEWICK, R. *Algorithms in C, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching*, 3ª edição. Addison-Wesley, 1997.

3. AED II: Conexões e Otimização (Grafos)



O Modelo Universal

- **Modelagem:** Redes Sociais, Rotas, Dependências de Software.
- **Otimização:** Encontrar o *melhor* caminho, não apenas *um* caminho.
- **Fronteira da Computação:** Lidar com problemas NP-Difíceis e algoritmos aproximados.

3. Estrutura dos Trabalhos Práticos

A disciplina é fortemente prática. Além das listas semanais (Juiz Online ou Questionários), teremos dois grandes **Trabalhos Práticos (TPs)** que consolidam os módulos:

TP1: Indexação e Estruturas Hierárquicas

- **Objetivo:** Implementar um sistema de índices (como um mini-Banco de Dados ou Buscador).
- **Conteúdo:** Árvores AVL, Rubro-Negra ou Árvores B.
- **Desafio:** Lidar com grandes volumes de dados onde a escolha da estrutura define se o programa roda em 1 segundo ou 1 hora.

TP2: Otimização em Grafos Realistas

- **Objetivo:** Resolver um problema de otimização em uma rede real (ex: Rotas de entregas, Alocação de horários, Rede de distribuição de água).
- **Conteúdo:** Caminhos Mínimos (Dijkstra), Árvore Geradora (Kruskal) ou Fluxo Máximo.
- **Desafio:** Modelagem. O difícil não é codar o Dijkstra, é perceber que o problema é um

- Exercícios semanais para prática intensiva.
- **Submissão Automática:** O sistema testa seu código contra diversos casos de teste ocultos.
- **Feedback Imediato:**
 - **Accepted (AC):** Código correto!
 - **Wrong Answer (WA):** Lógica incorreta.
 - **Time Limit Exceeded (TLE):** Algoritmo ineficiente.
- Mesmo modelo utilizado na **Maratona de Programação (SBC)**.



Figure: Logo Oficial Maratona de Programação (SBC)

4. Módulos da Disciplina e Objetivos (1/2)

Módulo 1: Fundamentos e Árvores de Pesquisa

- **Objetivos:** Revisar complexidade matemática, lidar com espalhamento (Hash) e introduzir a busca hierárquica em profundidade e largura (BST).

Módulo 2: Árvores Balanceadas e Memória Secundária

- **Objetivos:** Garantir a eficiência no pior caso (AVL, Rubro-Negra), explorar filas de prioridade (Heaps) e indexar dados massivos em disco (Árvores B/B+).

Módulo 3: Introdução e Percursos em Grafos

- **Objetivos:** Modelar problemas do mundo real em redes estruturadas e realizar percursos baseados em otimização de busca (BFS/DFS) e conexões de custo mínimo (Árvore Geradora Mínima).

4. Módulos da Disciplina e Objetivos (2/2)

Módulo 4: Caminhos Mínimos e Algoritmos Gulosos

- **Objetivos:** Resolver problemas de roteamento determinando rotas ótimas (Dijkstra, Bellman-Ford, Floyd-Warshall) e analisar cenários de alocação de recursos (Coloração de Grafos).

Módulo 5: Fluxo Máximo em Redes e Avançados

- **Objetivos:** Maximizar o transporte de itens através de gargalos (Ford-Fulkerson) e estudar classes de caminhos rigorosos (Grafos Eulerianos e Hamiltonianos).

<https://alessiojr.com/disciplines/aed2/>

Cronograma - Algoritmos e Estruturas de Dados II 2026-1

Apresentação

A disciplina de **Algoritmos e Estruturas de Dados II (AEDz)** do curso de **Engenharia de Computação** do **CEFET-MG** possui um histórico de evolução importante.

No currículo anterior (PPC 2008), a formação era composta por 4 aulas teóricas e 2 aulas de laboratório (Laboratório de AED2). Foi responsável por ministrar as ofetas de **2016.2 a 2021.2**, sempre buscando alinhar a base conceitual com a prática de implementação.

No atual **PPC 2023**, a disciplina de Laboratório foi descontinuada, mantendo-se a carga horária teórica. Para mitigar essa perda e garantir a excelência técnica, esta retomada da disciplina em **2026-1** incorpora uma abordagem prática integrada. A disciplina contará com **Exercícios Contínuos e Trabalhos Práticos** em **juizes online** (Maratonas de Programação), resgatando a experiência de “mão na massa” essencial para o domínio de **Estruturas de Árvores e Grafos**.

Conforme o plano de ensino vigente, a disciplina possui carga horária de **4 horas-aula semanais**. A avaliação é composta por **duas provas teóricas** (70 pontos no total), **dois trabalhos práticos** (14 pontos no total) e **exercícios contínuos** em formato de maratona (16 pontos no total).

Planos de Ensino 2026-1

Planos de Ensino 2026-1

- [Plano Didático \(PDF\)](#)
- [Plano de Ensino \(PDF\)](#)

Cronograma

As aulas ocorrem todas as **segundas-feiras às 07:00 e quartas-feiras às 10:40**

↳ Código

NESTA PÁGINA

Apresentação

Planos de Ensino 2026-1

Planos de Ensino 2026-1

Cronograma



log0-aed2

Aula	Data	Módulo	Conteúdo	Capítulo	Slides
------	------	--------	----------	----------	--------

5. Avaliação

Componente	Pts	Descrição
Prova 1	35	Árvores e Estruturas Hierárquicas
Prova 2	35	Grafos e Algoritmos
TP 1	7	Indexação (Árvores)
TP 2	7	Otimização (Grafos)
Listas	16	Juiz Online ou Questionários