

Plano da Disciplina: Algoritmos e Estrutura de Dados

1. Desenho da Disciplina

- **Nome:** Algoritmos e Estrutura de Dados
 - **Curso:** Licenciatura em Engenharia de Software, Ciência da Computação ou áreas correlatas
 - **Ano/Semestre:** 3º Ano / 2º Semestre
 - **Modalidade:** Presencial
 - **Carga Horária Semanal:** 5 horas
 - **Carga Horária Total:** 80 horas
-

2. Regime

- **Regime:** Semestral
 - **Frequência mínima:** 75% de presença nas aulas teórico-práticas
 - **Avaliação contínua** com provas, exercícios, projetos práticos e implementação de algoritmos
-

3. Posição no Curso

- **3º Ano, 2º Semestre**
 - Disciplina essencial para **desenvolvimento de estruturas de dados eficientes e algoritmos de manipulação de dados**
-

4. Horário Semanal

- **2 horas teóricas** (fundamentos e análise de algoritmos)
 - **1 hora teórico-prática** (exemplos aplicados e resolução de problemas computacionais)
 - **2 horas práticas (laboratórios)** (implementação e experimentação com algoritmos)
-

5. Carga Horária Total

- **Total:** 80 horas
 - **Horas Teóricas:** 32h
 - **Horas Teórico-Práticas:** 16h
 - **Horas Práticas (Laboratórios):** 32h

6. Precedências Obrigatórias

- **Fundamentos de Programação**
-

7. Objetivos Gerais

- 1 Desenvolver a **capacidade de representar e manipular listas lineares de forma sequencial e enlaçada**.
 - 2 Explorar **diferentes representações e operações sobre listas lineares, pilhas e filas**.
 - 3 Compreender a **estrutura de árvores e suas representações em memória**.
 - 4 Aplicar algoritmos eficientes de **ordenação e busca em estruturas de dados**.
 - 5 Implementar algoritmos de **representação e manipulação de grafos e suas aplicações**.
-

8. Competências a Desenvolver

- ✓ Representar e manipular **estruturas de dados fundamentais (listas, árvores, grafos, tabelas hash)**.
 - ✓ Aplicar técnicas de **busca e ordenação eficientes**.
 - ✓ Implementar **algoritmos recursivos e iterativos** para manipulação de dados.
 - ✓ Projetar **soluções computacionais otimizadas** com estruturas de dados avançadas.
 - ✓ Utilizar ferramentas computacionais como **Python, C, Java e bibliotecas especializadas**.
-

9. Conteúdo Temático e Bibliografia

Unidade 1: Tipos de Dados Abstratos e Estruturas Lineares (10h)

Brian W. Kernighan, Dennis Ritchie – The C Programming Language (1988)

- Definição de Tipos de Dados Abstratos (TAD)
- Listas lineares: estrutura sequencial e encadeada
- Implementação e manipulação de listas

Laboratório: Implementação de listas encadeadas e operações básicas

Unidade 2: Pilhas e Filas (10h)

Cairo & Guardati – Estruturas de Dados, McGraw-Hill (1994)

- Pilhas: conceitos, aplicações e implementação
- Filas e Filas Duplas (Deque)
- Aplicações de Pilhas e Filas

Laboratório: Implementação de pilhas e filas utilizando listas encadeadas

Unidade 3: Árvores e Recursão (15h)

Horowitz & Sahni – Fundamentals of Data Structures

- Estrutura e representação de árvores
- Árvores binárias: inserção, remoção e percursos
- Árvores binárias balanceadas

Laboratório: Implementação de árvores binárias de busca

Unidade 4: Algoritmos de Ordenação (15h)

Knuth – The Art of Computer Programming, Vol. 3: Sorting and Searching (1968)

- Ordenação por inserção, intercâmbio e seleção
- Algoritmos avançados: MergeSort, QuickSort e HeapSort
- Comparação de complexidade e otimizações

Laboratório: Implementação e comparação de algoritmos de ordenação

Unidade 5: Estruturas de Dados Não Lineares - Grafos (10h)

Robert Sedgewick – Algorithms in C (1997)

- Conceitos básicos de grafos
- Representação em listas de adjacências e matrizes
- Busca em largura (BFS) e profundidade (DFS)

Laboratório: Implementação de grafos e algoritmos BFS/DFS

Unidade 6: Tabelas Hash e Busca Avançada (10h)

Knuth – The Art of Computer Programming, Vol. 1: Fundamental Algorithms (1968)

- Tabelas Hash e funções de dispersão
- Manipulação de colisões
- Eficiência de busca

Laboratório: Implementação de uma estrutura Hash com tratamento de colisões

Unidade 7: Representação e Processamento de Estruturas em Memória (10h)

Fundamentos de Estruturas de Dados – Horowitz e Sahni

- Alocação dinâmica e estruturas encadeadas
- Uso de ponteiros e gerenciamento de memória
- Representação de listas em memória externa

Laboratório: Implementação de uma estrutura de dados dinâmica

Unidade 8: Projeto Final Integrado (10h)

Baseado nas competências adquiridas ao longo do semestre

- Desenvolvimento de um **projeto prático aplicando estruturas de dados e algoritmos avançados**
- Modelagem e análise de desempenho

Laboratório: Desenvolvimento e defesa do projeto final

10. Metodologia de Trabalho

Aulas teóricas para introdução aos conceitos fundamentais de estruturas de dados

Exercícios práticos para reforço da implementação de algoritmos

Laboratórios computacionais para experimentação e otimização de código

Projetos individuais e em grupo para aprofundamento dos conceitos

11. Critério de Avaliação

- ✓ **40% - Trabalhos práticos e exercícios laboratoriais**
 - ✓ **30% - Projeto Final Integrado**
 - ✓ **20% - Participação e atividades complementares**
 - ✓ **10% - Prova escrita e técnica**
-

12. Bibliografia Básica

Brian W. Kernighan, Dennis Ritchie – *The C Programming Language*, Prentice Hall (1988)

Cairo / Guardati – *Estruturas de Dados*, McGraw-Hill (1994)

Horowitz & Sahni – *Fundamentals of Data Structures*

Knuth, D. – *The Art of Computer Programming, Vol. 1 & Vol. 3*, Addison Wesley (1968)

Robert Sedgewick – *Algorithms in C*, Addison-Wesley (1997)