

Plano Analítico: Programação I (Análise de Algoritmos e Estruturas de Dados)

1. Identificação da Unidade Curricular

- **Instituição:** Instituto Superior Politécnico de Ciências e Tecnologia (INSUTEC)
- **Curso:** Engenharia de Informática e Sistemas de Informação (EISI)
- **Classificação:** Disciplina Específica (Nuclear)
- **Ano:** 1º | **Semestre:** 2º (2º Semestre)
- **Créditos:** 8.0 UC
- **Carga Horária Total:** 120 Horas (90h de Contacto | 30h de Trabalho Complementar)

2. Apresentação e Justificação

Esta Unidade Curricular é o pilar da engenharia de software eficiente. O foco recai sobre a análise da complexidade de algoritmos e a escolha da estrutura de dados ideal (Listas, Pilhas, Filas, Árvores) para otimizar o tempo de execução e o uso de memória. A disciplina justifica-se pela necessidade de formar engenheiros capazes de criar sistemas escaláveis, fundamentados em princípios de computação científica e boas práticas de programação em C/C++, em linha com o **Decreto Presidencial 193/18**.

3. Competências a Desenvolver (Decreto 193/18)

3.1 Competências Instrumentais (Saber)

- Compreender a notação **Big-O** e a análise de complexidade temporal e espacial.
- Dominar as estruturas de dados lineares e não-lineares.
- Entender algoritmos de ordenação e pesquisa e os seus limites de performance.

3.2 Competências Técnicas e Operacionais (Saber Fazer)

- **Implementação de Estruturas:** Desenvolver Listas Ligadas, Pilhas e Filas utilizando alocação dinâmica de memória.
- **Otimização:** Selecionar e adaptar algoritmos de ordenação (QuickSort, MergeSort) para casos específicos.
- **Gestão de Memória:** Manipular ponteiros e endereçamento para criação de estruturas dinâmicas complexas.

3.3 Competências Atitudinais (Saber Ser/Estar)

- Demonstrar rigor analítico na comparação de diferentes soluções algorítmicas.
- Valorizar o equilíbrio entre a simplicidade do código e a eficiência de execução.

4. Conteúdo Temático (Estrutura de 120 Horas)

1. **Introdução à Análise de Algoritmos:** Notação assintótica (Big-O, Omega, Theta) e contagem de operações.
2. **Recursividade:** Conceitos, funcionamento da pilha de execução e análise de recorrências.

3. **Algoritmos de Ordenação e Pesquisa:** Bubble, Insertion, Selection vs. QuickSort e MergeSort; Pesquisa Binária.
4. **Estruturas de Dados Lineares Dinâmicas:** Listas Simplesmente e Duplamente Ligadas; Pilhas (Stacks) e Filas (Queues).
5. **Gestão Dinâmica de Memória:** Ponteiros, malloc, free e prevenção de fugas de memória (*memory leaks*).
6. **Tipos Abstratos de Dados (TADs):** Encapsulamento e modularização de estruturas.
7. **Introdução a Estruturas Não-Lineares:** Árvores Binárias de Pesquisa (BST) e conceitos básicos de Grafos.

5. Regime de Avaliação (Disciplina Específica)

- **Avaliação Contínua (40%):**
 - 1ª Frequência (Análise de Complexidade e Ordenação): 13%
 - 2ª Frequência (Estruturas Dinâmicas e TADs): 14%
 - **Projecto de Implementação:** Biblioteca de Estruturas de Dados em C: 13%
- **Exame Normal (60%):** Prova global teórica e prática (resolução de problemas algorítmicos).

6. Referências Bibliográficas (APA 7ª Ed.)

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms* (4th ed.). MIT Press.
- Sedgwick, R., & Wayne, K. (2011). *Algorithms* (4th ed.). Addison-Wesley.
- Tenenbaum, A. M. (1995). *Estruturas de dados utilizando C*. Makron Books.
- Pinheiro, M. (2012). *Algoritmos e estruturas de dados*. LTC.