

Plano da Disciplina: Fundamentos de Programação

1. Desenho da Disciplina

- **Nome:** Fundamentos de Programação
 - **Curso:** Licenciatura em Ciência da Computação, Engenharia de Software ou áreas correlatas
 - **Ano/Semestre:** 2º Ano / 1º 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 exercícios, projetos práticos e provas
-

3. Posição no Curso

- **2º Ano, 1º Semestre**
 - Disciplina fundamental para **bases de programação estruturada e modular**
-

4. Horário Semanal

- **2 horas teóricas** (conceitos de algoritmos e programação)
 - **1 hora teórico-prática** (exemplos aplicados e desenvolvimento de código)
 - **2 horas práticas (laboratórios)** (implementação de programas e depuração)
-

5. Carga Horária Total

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

6. Precedências Obrigatórias

- **Não há disciplinas precedentes obrigatórias.**
-

7. Objetivos Gerais

- 1 Fornecer **fundamentos da lógica de programação e estruturação de algoritmos.**
 - 2 Ensinar **tipos de dados, operadores e estruturas de decisão e repetição.**
 - 3 Capacitar os alunos para **desenvolvimento de programas estruturados e modulares.**
 - 4 Introduzir conceitos básicos de **vetores, matrizes e manipulação de dados.**
 - 5 Preparar os alunos para linguagens de programação de alto nível como **C, C++ e Java.**
-

8. Competências a Desenvolver

- ✓ Compreender e aplicar **estruturas fundamentais de programação.**
 - ✓ Desenvolver e otimizar **algoritmos eficientes para resolução de problemas computacionais.**
 - ✓ Escrever **programas modulares e reutilizáveis** com procedimentos e funções.
 - ✓ Utilizar **vetores e matrizes** na manipulação de dados estruturados.
 - ✓ Depurar e testar programas para **garantir sua correta execução.**
-

9. Conteúdo Temático e Bibliografia

Unidade 1: Lógica de Programação e Introdução a Algoritmos (10h)

Forbellone & Eberspächer – Lógica de Programação (2005)

- O que é um algoritmo?
- Estruturas de controle básicas
- Pseudocódigo e fluxogramas

Laboratório: Implementação de primeiros algoritmos com pseudocódigo

Unidade 2: Tipos de Dados e Operações (10h)

Deitel & Deitel – C: Como Programar (2016)

- Variáveis, constantes e escopo
- Tipos de dados primitivos e estruturados
- Operadores aritméticos, relacionais e lógicos

Laboratório: Implementação de cálculos e operações com variáveis

Unidade 3: Estruturas Condicionais (15h)

Ascencio & Campos – Fundamentos da Programação de Computadores (2016)

- Estrutura **se-então**
- Estrutura **se-então-senão**
- Estrutura **escolha-caso**

Laboratório: Simulação de fluxos de controle em programas simples

Unidade 4: Estruturas de Repetição (15h)

Schildt – C++: The Complete Reference (2014)

- Estrutura **para (for)**
- Estrutura **enquanto (while)**
- Estrutura **repita (do-while)**

Laboratório: Implementação de laços de repetição para processamento de dados

Unidade 5: Vetores e Matrizes (10h)

Ascencio & Campos – Fundamentos da Programação de Computadores (2016)

- Declaração e inicialização de vetores
- Aplicação de vetores para armazenamento e busca
- Declaração e manipulação de matrizes

Laboratório: Desenvolvimento de algoritmos para manipulação de listas e tabelas

Unidade 6: Programação Modular (10h)

Deitel & Deitel – C: Como Programar (2016)

- Definição de **procedimentos e funções**
- Passagem de parâmetros e retorno de valores
- Modularização e reutilização de código

Laboratório: Implementação de funções e bibliotecas personalizadas

Unidade 7: Projeto Final Integrado (10h)

Baseado nas competências adquiridas ao longo do semestre

- Desenvolvimento de um **projeto prático de programação**
- Aplicação dos conceitos estudados para resolver um problema computacional

Laboratório: Implementação e defesa do projeto final

10. Metodologia de Trabalho

Aulas teóricas para introdução aos conceitos fundamentais de programação

Exercícios práticos para reforço da lógica computacional

Laboratórios computacionais para experimentação e depuração de código

Projetos individuais e em grupo para aplicação prática dos conceitos estudados

11. Critério de Avaliação

✓ **40% - Trabalhos práticos e exercícios laboratoriais**

✓ **30% - Projeto Final Integrado**

✓ **20% - Participação e atividades complementares**

✓ **10% - Prova escrita/técnica**

12. Bibliografia Básica

Deitel, P. J., & Deitel, H. M. - *C: Como Programar*, 8ª ed., Pearson (2016)

Forbellone, A. L. V., & Eberspächer, H. F. - *Lógica de Programação*, 3ª ed., Pearson (2005)

Ascencio, A. & Campos, E. - *Fundamentos da Programação de Computadores*, 3ª ed., Pearson (2016)

Schildt, H. - *C++: The Complete Reference*, 4ª ed., McGraw-Hill (2014)

13. Bibliografia Complementar

Sedgewick, R., & Wayne, K. - *Computer Science: An Interdisciplinary Approach*, Addison-Wesley (2017)

Horstmann, C. S. - *Core Java, Volume I - Fundamentals*, 11ª ed., Prentice Hall (2019)

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. - *Algoritmos: Teoria e Prática*, 3ª ed., Elsevier (2009)

Zelle, J. M. - *Python Programming: An Introduction to Computer Science*, Franklin, Beedle & Associates (2010)

Kernighan, B. W., & Ritchie, D. M. - *The C Programming Language*, 2ª ed., Prentice Hall (1988)