

PLANO ANALÍTICO - INFORMÁTICA E SISTEMAS DE INFORMAÇÃO

1. Designação da Disciplina:

INTRODUÇÃO AOS COMPUTADORES E À PROGRAMAÇÃO

2. Regime:

Semestral

3. Posição no Curso:

1º ano, 2º semestre

4. Horário Semanal:

- **5 horas semanais**, distribuídas em:
 - o 2 horas teóricas
 - o 2 horas teórico-práticas
 - o 1 hora prática (laboratório)

5. Carga Horária Total:

- **80 horas semestrais**, distribuídas em:
 - o **Horas teóricas:** 32h
 - o **Horas teórico-práticas:** 32h
 - o **Horas práticas (laboratório):** 16h

6. Precedências Obrigatórias:

Não tem

7. Objetivos Gerais:

- Introduzir conceitos fundamentais de programação estruturada.
- Desenvolver a lógica de programação e a construção de algoritmos eficientes.
- Ensinar o uso de variáveis, operadores, estruturas de controle e repetição.
- Capacitar os alunos a desenvolver programas em linguagens de alto nível como **C, C++ ou Java**.

8. Competências a Desenvolver:

Ao final da disciplina, os estudantes deverão ser capazes de:

- Compreender e aplicar os fundamentos da lógica de programação.
- Criar algoritmos estruturados para resolução de problemas computacionais.
- Utilizar corretamente variáveis, operadores e estruturas de controle de fluxo.
- Implementar programas funcionais em C, C++ ou Java.

- Modularizar código utilizando funções e procedimentos.
-

9. Conteúdo Temático e Bibliografia

UNIDADE I - Lógica de Programação e Algoritmos (16h)

Tópicos:

- 1.1. Fundamentos da lógica de programação
- 1.2. Algoritmos e fluxogramas
- 1.3. Introdução às linguagens de programação
- 1.4. Pseudocódigo e boas práticas na escrita de algoritmos

Laboratório:

- Implementação de algoritmos básicos usando pseudocódigo.
- Uso de ferramentas de simulação de algoritmos (como VisuAlg).

Bibliografia:

- Forbellone, A. L. V.; Eberspächer, H. F., *Lógica de Programação: A Construção de Algoritmos e Estruturas de Dados*, Pearson, 2005.
 - Ascencio, A.; Campos, E., *Fundamentos da Programação de Computadores*, Pearson, 2016.
-

UNIDADE II - Tipos de Dados e Operadores (16h)

Tópicos:

- 2.1. Tipos de dados: inteiros, reais, caracteres e booleanos
- 2.2. Variáveis e constantes
- 2.3. Operadores aritméticos, relacionais e lógicos
- 2.4. Precedência de operadores

Laboratório:

- Exercícios práticos de declaração e manipulação de variáveis.
- Cálculos com operadores aritméticos, relacionais e lógicos em C/C++.

Bibliografia:

- Deitel, P. J.; Deitel, H. M., *C: Como Programar*, Pearson, 2016.
 - Kernighan, B. W.; Ritchie, D. M., *The C Programming Language*, Prentice Hall, 1988.
-

UNIDADE III - Estruturas de Controle de Fluxo (16h)

Tópicos:

- 3.1. Estruturas condicionais:

- if-else

- switch-case
- 3.2. Estruturas de repetição:
- for
 - while
 - do-while

Laboratório:

- Implementação de algoritmos com estrutura condicional e laços de repetição.
- Exercícios práticos de controle de fluxo.

Bibliografia:

- Schildt, H., *C++: The Complete Reference*, McGraw-Hill, 2014.
 - Zelle, J. M., *Python Programming: An Introduction to Computer Science*, Franklin, Beedle & Associates Inc., 2010.
-

UNIDADE IV - Estruturas de Dados Simples (16h)

Tópicos:

- 4.1. Vetores e sua aplicação
- 4.2. Matrizes e manipulação de dados bidimensionais

Laboratório:

- Implementação de programas que utilizam vetores e matrizes.
- Manipulação de listas e arrays em C/C++.

Bibliografia:

- Deitel, P. J.; Deitel, H. M., *C: Como Programar*, Pearson, 2016.
 - Horstmann, C. S., *Core Java, Volume I - Fundamentals*, Prentice Hall, 2019.
-

UNIDADE V - Programação Modular e Funções (16h)

Tópicos:

- 5.1. Definição e uso de funções e procedimentos
- 5.2. Passagem de parâmetros e escopo de variáveis
- 5.3. Modularização de código e boas práticas

Laboratório:

- Implementação de programas modulares usando funções em C e Java.
- Exercícios práticos sobre escopo e reutilização de código.

Bibliografia:

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

- Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C., *Algoritmos: Teoria e Prática*, Elsevier, 2009.
-

10. Metodologia de Trabalho

- **Aulas teóricas:** Explicação dos fundamentos da programação e algoritmos.
 - **Aulas teórico-práticas:** Resolução de exercícios e estudo de casos reais.
 - **Laboratórios:** Desenvolvimento de programas e simulação de algoritmos em C, C++ e Java.
-

11. Critério de Avaliação

A avaliação será realizada da seguinte forma:

- **Testes escritos (40%)** – avaliação teórica sobre os conceitos de programação.
 - **Projetos práticos (30%)** – desenvolvimento de pequenos programas aplicando os conteúdos estudados.
 - **Exame final (30%)** – avaliação geral do conteúdo do curso.
-

12. Bibliografia Básica

- Deitel, P. J.; Deitel, H. M., *C: Como Programar*, Pearson, 2016.
 - Forbellone, A. L. V.; Eberspächer, H. F., *Lógica de Programação: A Construção de Algoritmos e Estruturas de Dados*, Pearson, 2005.
 - Ascencio, A.; Campos, E., *Fundamentos da Programação de Computadores*, Pearson, 2016.
-

13. Bibliografia Complementar

- Sedgewick, R.; Wayne, K., *Computer Science: An Interdisciplinary Approach*, Addison-Wesley, 2017.
- Horstmann, C. S., *Core Java, Volume I – Fundamentals*, Prentice Hall, 2019.
- Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C., *Algoritmos: Teoria e Prática*, Elsevier, 2009.
- Zelle, J. M., *Python Programming: An Introduction to Computer Science*, Franklin, Beedle & Associates Inc., 2010.