



MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

**PLANO DE ENSINO**

---

**1. Identificação**

**Disciplina:** INE5402 - Programação Orientada a Objetos I

**Nível:** Graduação

**Carga Horária:** 108 horas-aula (Teórica: 30; Prática: 78)

**Período Letivo:** 2026-1

---

**Turma:** 01208B - CIÊNCIAS DA COMPUTAÇÃO (208)  
Horário: 2 13:30 3 (CTC301), 4 15:10 3 (INE101)  
Docente: Maicon Rafael Zatelli (maicon.zatelli@ufsc.br)

---

**2. Ementa**

Algoritmo. Programação de computador. Resolução de problemas usando computador. Linguagem de Programação. Elementos de programação imperativa: variáveis simples, coleções uni e n dimensionais; tipos das variáveis; comandos; procedimentos e funções. Conceitos fundamentais do paradigma de Programação Orientada a Objetos: objeto, atributo, método, classe. Prática de programação usando alguma linguagem de programação orientada a objetos.

---

**3. Objetivos**

**3.1 Objetivo Geral:**

Apresentar as noções básicas de programação de computadores capacitando os alunos a analisar problemas de complexidade básica e projetar/desenvolver soluções de software sob a perspectiva de orientação a objetos.

**3.2 Objetivos Específicos:**

- a) Apresentar os conceitos fundamentais da programação orientada a objetos.
  - b) Capacitar o aluno a analisar problemas de complexidade básica, abstraindo, modelando e implementando soluções sob o enfoque de programação orientada a objetos.
  - c) Desenvolver fluência em uma linguagem de programação orientada a objetos.
- 

**4. Conteúdo Programático**

1 CONTEXTUALIZAÇÃO [8 horas-aula]

1.1 Modelo conceitual

1.2 Processos de abstração e representação

1.3 Histórico sobre linguagens de programação

2 CONCEITOS BÁSICOS DA ORIENTAÇÃO A OBJETOS [10 horas-aula]

- 2.1 Classes e objetos
- 2.2 Atributos
- 2.3 Métodos, argumentos e parâmetros
- 3 CONCEITOS BÁSICOS DE PROGRAMAÇÃO IMPERATIVA [46 horas-aula]
  - 3.1 Algoritmos e programas
  - 3.2 Processo de edição, compilação e execução
  - 3.3 Variáveis e Tipos de dados
  - 3.4 Comando de atribuição
  - 3.5 Operadores aritméticos e lógicos
  - 3.6 Estruturas de controle
    - 3.6.1 Estrutura de seqüenciação
    - 3.6.2 Estruturas de decisão (simples e compostas)
    - 3.6.3 Estruturas de repetição (condicionais e contadas)
- 4 COLEÇÕES [36 horas-aula]
  - 4.1 Cadeias de caracteres (String)
  - 4.2 Coleções unidimensionais (Array, Lista, Tupla)
  - 4.3 Coleções bidimensionais (Matriz)
  - 4.4 Coleções indexadas (Dicionário)
- 5 MODELO DE OBJETOS [08 horas-aula]
  - 5.1 Comunicação por troca de mensagens
  - 5.2 Encapsulamento e ocultamento de informações
  - 5.3 Hierarquia de agregação/decomposição
  - 5.4 Hierarquia de especialização/generalização
  - 5.5 Herança e Polimorfismo

---

## 5. Metodologia

O conteúdo teórico será apresentado em sala por meio de aulas expositivas e dialogadas, onde conceitos, técnicas de programação, problemas e soluções serão discutidos. Aulas práticas em laboratório consistirão na resolução de listas de exercícios de programação para fixação do conteúdo. A quantidade de listas de exercícios será em torno de 10, distribuídos ao longo do semestre. É esperado que cada estudante envolva-se ativamente na resolução das listas de exercícios propostas. As possíveis soluções para alguns dos exercícios das listas serão discutidos durante as aulas no laboratório.

---

## 6. Avaliação de Aprendizagem

Será considerado aprovado na disciplina o aluno que obtiver Média Final (MF) igual ou superior a 6,0 e frequência igual ou superior a 75%.

A avaliação será feita por meio de 5 provas (P1, P2, P3, P4, P5) e 1 trabalho prático (TP).

As provas consistirão na resolução de uma lista de problemas, onde o aluno deverá escrever programas ou partes de programas para resolvê-los.

O trabalho prático poderá ser feito em grupo e consistirá no desenvolvimento de um programa de maior complexidade que utilize os conceitos práticos trabalhados no semestre.

A média final (MF) de cada estudante é calculada pela fórmula abaixo:

$$MF = P1 * 0,1 + P2 * 0,2 + P3 * 0,2 + P4 * 0,2 + P5 * 0,2 + TP * 0,1$$

---

## 7. Recuperação

Dado que a disciplina apresenta pelo menos 50% da carga horária consistindo de aulas práticas, conforme deliberação do Colegiado do Curso de Ciências da Computação de 18 de março de 2008, ela não prevê a realização de avaliação no final do semestre (recuperação) de que trata o parágrafo 2º do artigo 70 da Resolução 17/CUn/97.

---

## 8. Cronograma

O cronograma da disciplina segue a ordenação dos tópicos descritos no programa. As provas serão realizadas, aproximadamente, ao final de cada mês do semestre. As datas exatas serão divulgadas com antecedência no Moodle. O enunciado do trabalho será divulgado aproximadamente na sexta semana do semestre. Por fim, exercícios serão distribuídos ao longo do semestre a cada novo conteúdo apresentado.

---

## 9. Bibliografia Básica

- [1] SEVERANCE, Charles R. Python para Todos. Publicação Independente, 2020. Disponível em <http://do1.dr-chuck.com/pythonlearn>
  - [2] WAZLAWICK, Raul S. Introdução a Algoritmos e Programação com Python. São Paulo: Elsevier, 2017.
  - [3] OLIVEIRA, Jayr F. e MANZANO, José Augusto N. G. Algoritmos. Érica, 2016.
  - [4] SOUZA, Marco F. de Souza; et al. Algoritmos e Lógica de Programação. São Paulo: Thomson Learning, 2005.
- 

## 10. Bibliografia Complementar

- [1] BOOCH, G., Object-Oriented Design. Benjamin/Cummings Pub. 1998.
- [2] MEYER, Bertrand. Object-oriented software construction. 2nd. ed. Upper Saddle River, NJ: Prentice-Hall PTR, 1997.
- [3] RUMBAUGH, James et alii. Modelagem e Projetos Baseados em Objetos. Ed. Campus, 1994.
- [4] WAZLAWICK, Raul S. Análise e Projeto de Sistemas de Informação Orientados a Objetos. São Paulo: Campus. 2004.