



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:** 01208C - CIÊNCIAS DA COMPUTAÇÃO (208)  
Horário: 2 15:10 3 (LIICT3), 4 15:10 3 (LIICT3)  
Docente: Antonio Carlos Mariani (antonio.c.mariani@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 da disciplina é dividido numa seqüência de módulos disponíveis no ambiente Moodle, sendo que há exercícios práticos a serem desenvolvidos pelos estudantes, totalizando em torno de 130 exercícios. Os módulos são:

1. Tartaruga (turtle)
2. Estrutura de Sequenciação
3. Estrutura de Seleção
4. Estruturas de Repetição (iteração)
5. Cadeia de caracteres (String)
6. Tuplas, listas, conjuntos e dicionários
7. Coleções Bidimensionais (matrizes)
8. Orientação a Objetos

Os aspectos teóricos da disciplina serão abordados ao longo do semestre em aulas expositivas, sempre precedidas de leitura de material correspondente disponibilizado nos ambientes Moodle UFSC.

A prática de programação será desenvolvida utilizando a linguagem de programação Python por meio de implementação computacional de soluções para problemas propostos, os quais estarão referenciados no Moodle em cada um dos módulos através de atividades do tipo VPL (Laboratório Virtual de Programação). Na resolução dos problemas o estudante pode escolher o ambiente de programação Python que lhe seja mais apropriado, com preferência para o ambiente Thonny (o qual é particularmente indicado para iniciantes em programação de

computadores) ou na própria atividade VPL do Moodle. Os trabalhos práticos poderão ser desenvolvidos individualmente ou em grupo, mas a postagem no Moodle, de caráter opcional, é individual, devendo ser feita via atividades VPL do Moodle.

A resolução de atividades práticas, assim como orientações e discussões, ocorrerão ao longo do semestre através de encontros presenciais contínuos em laboratório de informática a ser disponibilizado pelo Departamento de Informática e Estatística ou pelo Centro Tecnológico. Nesses encontros serão também parcial ou totalmente resolvidos problemas dentre aqueles que compõe as atividades práticas da disciplina, em particular aqueles indicados por estudante.

### **Controle de frequência**

O registro de frequência será feito via módulo de registro de frequência do Moodle, propiciando gerenciamento e controle pelos estudantes diretamente no ambiente Moodle.

---

## **6. Avaliação de Aprendizagem**

Satisfeito o critério de 75% de frequência, a avaliação progressiva da disciplina será feita por meio de três provas presenciais.

Considerando que os conteúdos dos módulos acima descritos são progressivamente um pré-requisito do outro, as provas contemplarão:

- P1: os módulos de 2 a 4;
- P2: os módulos de 2 a 6;
- P3: todos os módulos.

As provas são individuais, consistindo da resolução de uma lista de problemas propostos, nos mesmos moldes dos exercícios práticos. As provas serão feitas em laboratórios de informática utilizando o Quiosque de Provas, sendo necessário haver um computador por aluno para sua realização.

A média final da disciplina (MF) é dada por:  $MF = (P1 \times 2 + P2 \times 3 + P3 \times 4) / 9$

---

## **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**

A tabela abaixo apresenta as previsões de apresentação de conteúdos e desenvolvimento contínuo das atividades práticas.

Módulo	Previsão
1. Tartaruga (turtle)	1,5 a 2 semanas
2. Estrutura de Sequenciação	1,5 a 2 semanas
3. Estrutura de Seleção	2 a 2,5 semanas
4. Estruturas de Repetição (iteração)	2 a 2,5 semanas
5. Cadeia de caracteres (String)	1 semana
6. Tuplas, listas, conjuntos e dicionários	2 a 3 semanas
7. Coleções Bidimensionais (matrizes)	2 semanas
8. Orientação a Objetos	1 a 2 semanas

As provas ocorrerão com o espaçamento mínimo de uma semana após o término do último módulo englobado, conforme abaixo:

Prova	Data Prevista
P1	11/05/2026
P2	16/06/2026
P3	06/07/2026

---

## 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.