



Plano de Ensino

1) Identificação

Disciplina:	INE5603 - Introdução à Programação Orientada a Objetos		
Turma(s):	01238A		
Carga horária:	108 horas-aula	Teóricas: 30	Práticas: 78
Período:	1º semestre de 2025		

2) Cursos

- Sistemas de Informação (238)

3) Requisitos

- Não há

4) Professores

- Antonio Carlos Mariani (antonio.c.mariani@ufsc.br)

5) Ementa

Modelagem conceitual: Abstração X Representação. O Modelo de Objetos: Classes e Objetos, Comunicação por troca de mensagens. Herança e Polimorfismo.

6) Objetivos

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.

Específicos:

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

7) Conteúdo Programático

- 7.1) CONTEXTUALIZAÇÃO [8 horas-aula]
 - Modelo conceitual
 - Processos de abstração e representação
 - Histórico sobre linguagens de programação
- 7.2) CONCEITOS BÁSICOS DA ORIENTAÇÃO A OBJETOS [10 horas-aula]
 - Classes e objetos
 - Atributos
 - Métodos, argumentos e parâmetros
- 7.3) CONCEITOS BÁSICOS DE PROGRAMAÇÃO IMPERATIVA [46 horas-aula]
 - Algoritmos e programas
 - Processo de edição, compilação e execução
 - Variáveis e Tipos de dados
 - Comando de atribuição
 - Operadores aritméticos e lógicos
 - Estruturas de controle
 - Estrutura de seqüenciação
 - Estruturas de decisão (simples e compostas)
 - Estruturas de repetição (condicionais e contadas)
- 7.4) COLEÇÕES [36 horas-aula]
 - Cadeias de caracteres (String)
 - Coleções unidimensionais (Array, Lista, Tupla)

- Coleções bidimensionais (Matriz)
- Coleções indexadas (Dicionário)

7.5) MODELO DE OBJETOS [08 horas-aula]

- Comunicação por troca de mensagens
- Encapsulamento e ocultamento de informações
- Hierarquia de agregação/decomposição
- Hierarquia de especialização/generalização
- Herança e Polimorfismo

8) Metodologia

O conteúdo da disciplina é dividido numa sequência 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:

- Tartaruga (turtle)
- Estrutura de Sequenciação
- Estrutura de Seleção
- Estruturas de Repetição (iteração)
- Cadeia de caracteres (String)
- Tuplas, listas, conjuntos e dicionários
- Coleções Bidimensionais (matrizes)
- 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 e/ou Jupyter Notebooks, este último referenciado no Moodle.

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. Nestes 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.

9) Avaliação

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, a primeira (P1) contemplará os módulos de 2 a 4, a segunda (P2) os módulos de 2 a 6 e a terceira (P3) incluirá todos os módulos.

As provas são individuais, consistindo da resolução de uma lista de problemas propostos, nos mesmos moldes dos exercícios práticos.

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

Não é prevista atividade de recuperação para esta turma, nos termos previstos no art. 70, parágrafo 2o, da Resolução 17/CUn/97, uma vez que cumpre pelo menos um dos seguintes requisitos:

- ter pelo menos 50% de carga prática;
- ter pelo menos 50% do peso da média final originado de trabalho prático;
- ter a inadequação da aplicação de avaliação de recuperação reconhecida pelo colegiado do curso, a partir da avaliação de solicitação fundamentada de dispensa de avaliação de recuperação, encaminhada pelo(s) professor(es) autor(es) do respectivo plano de ensino, para disciplinas com carga prática prevista no programa da disciplina, com nota de trabalho prático considerada no cálculo da média final e que não tenham cumprido um dos requisitos anteriores.

10) Cronograma

Há uma previsão de entorno de 1,5 a 2 semanas para cada um dos módulos descritos anteriormente, com desenvolvimento contínuo das atividades práticas. As provas ocorrerão com o espaçamento mínimo de uma semana após o término do último módulo englobado.

11) Bibliografia Básica

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

12) Bibliografia Complementar

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