



**UNIVERSIDADE FEDERAL DE SANTA CATARINA – UFSC**  
CENTRO DE CIÊNCIAS DA EDUCAÇÃO – CED  
DEPARTAMENTO DE CIÊNCIA DA INFORMAÇÃO – CIN  
CAMPUS UNIVERSITÁRIO- TRINDADE  
CEP: 88040-970 - FLORIANÓPOLIS - SANTA CATARINA  
Fone: (048) 3721-4075 Email: cin@contato.ufsc.br

## **PLANO DE ENSINO**

### **1 IDENTIFICAÇÃO**

Disciplina: **CIN7907 - Lógica Aplicada I**

Carga Horária: 72 H/A - 4 créditos

Oferta: Optativa para os Cursos de Graduação em Ciência da Informação, Biblioteconomia, Arquivologia e áreas afins.

Horário: Sextas-feiras das 18:30 às 22:00 h.

Local: a definir.

### **EMENTA**

Modelagem de sistemas básicos. Estruturas de dados simples. Algoritmos. Orientação a objetos.

### **2 OBJETIVOS**

#### **2.1 Objetivo Geral**

Desenvolver a capacidade de resolução de problemas lógicos de complexidade básica por meio da construção e aplicação de algoritmos.

#### **2.2 Objetivos Específicos**

2.2.1 Identificar elementos de problemas lógicos;

2.2.2 Modelar cenários de aplicação orientados a objetos;

2.2.3 Modelar estruturas de dados de suporte a algoritmos;

2.2.4 Implementar soluções algorítmicas a partir do paradigma da orientação a objetos.

### **3 CONTEÚDO PROGRAMÁTICO**

#### **3.1 Máquina de Turing**

#### **3.2 Arquiteturas Lógicas de Processamento Computacional**

3.2.1 Arquitetura de John Von Neumann

3.2.1.1 Unidade de Controle (UC)

3.2.1.2 Memória

3.2.1.3 Unidade Lógica e Aritmética (ULA)

3.2.1.4 Dispositivos de Entrada e Saída de Dados (E/S)

3.2.2 Arquitetura de Harvard

3.2.3 Outras Arquiteturas

#### **3.3 Algoritmos**

3.3.1 Histórico

3.3.2 Tipos de Dados

3.3.3 Variáveis Simples

- 3.3.4 Variáveis Compostas
- 3.3.5 Procedimentos
- 3.3.6 Funções
- 3.3.7 Condições
- 3.3.8 Laços de Repetição

### 3.4 Orientação a Objetos

- 3.4.1 Objeto
- 3.4.2 Atributo
- 3.4.3 Método
- 3.4.4 Classe
- 3.4.5 Herança
- 3.4.6 Polimorfismo

### 3.5 Resolução de Problemas Lógicos via Algoritmos

## 4 BIBLIOGRAFIAS

### 4.1 Bibliografia Básica

ARAÚJO, Everton Coimbra de. **Algoritmos: fundamento e prática**. 3. ed. ampl. e atual. Florianópolis: Visual Books, 2007. 414 p. ISBN 9788575022092.

GUIMARÃES, Angelo de Moura; LAGES, Newton Alberto de Castilho. **Algoritmos e estruturas de dados**. Rio de Janeiro: Livros Técnicos e Científicos, 1985. 216p. ISBN 8521603789.

PREISS, Bruno R. **Estruturas de dados e algoritmos: padrões de projetos orientados a objetos com java**. Rio de Janeiro: Campus, c2001. xvi, 566p. ISBN 8535206930.

### 4.2 Bibliografia Complementar

BOOCH, Grady et al. **Object-oriented analysis and design with applications**. 3rd ed. Upper Saddle River: Addison Wesley, c2007. xxiii, 691 p. ISBN 9780201895513.

BLAHA, Michael; RUMBAUGH, James. **Modelagem e projetos baseados em objetos com UML 2**. 2. ed. rev. e atual. Rio de Janeiro: Elsevier, 2006. xvii, 496 p. ISBN 8535217533.

HEEGE, Marcus. **Expert C++/CLI: .NET for Visual C++ Programmers**. Berkeley, CA: Apress, Inc., 2007. ISBN 9781430203575 Disponível em: <http://dx.doi.org/10.1007/978-1-4302-0357-5>. Acesso em: 14 jul. 2022.

HETLAND, Magnus Lie. **Beginning Python: From Novice to Professional**. Berkeley, CA: Apress, Inc., 2005. ISBN 9781430200727 Disponível em: <http://dx.doi.org/10.1007/978-1-4302-0072-7>. Acesso em: 14 jul.2022.

LANGTANGEN, Hans Petter. **Python Scripting for Computational Science**. Third Edition. Berlin: Springer-Verlag Berlin Heidelberg, 2008. (Texts in Computational Science and Engineering, 1611-0994; 3).

LARMAN, Craig. **Applying UML and patterns: an introduction to object-oriented analysis and design and the unified process**. 2nd ed. Upper Saddle River: Prentice Hall PTR, c2002. xxi, 627 p. ISBN 0130925691.

MEDEIROS, Ernani Sales de. **Desenvolvendo software com UML 2.0**: definitivo. São Paulo: Pearson Makron Books, 2004. 264 p. ISBN 8534615292.

RANGEL, Ricardo. **Curso prático: programação orientada a objetos**: com Borland C++ para Windows. São José dos Campos: Rangel, 1998. 473p.

SILVA, Ricardo Pereira e. **Como modelar com UML 2**. Florianópolis: Visual Books, 2009. 319p. ISBN 9788575022436.

STROUSTRUP, Bjarne. **A linguagem de programação C++**. 3. ed. Porto Alegre: Bookman, 2000. 823p. ISBN 8573076992.

WAZLAWICK, Raul Sidnei. **Introdução a algoritmos e programação com Python**: uma abordagem dirigida por testes. Rio de Janeiro: Elsevier, 2018. 214 p. ISBN 9788535290516.

## 5 METODOLOGIA E INSTRUMENTOS DE AVALIAÇÃO

Aulas expositivas e dialogadas; trabalhos práticos e avaliações. A verificação do rendimento escolar compreenderá frequência e aproveitamento nos estudos, os quais deverão ser atingidos conjuntamente. Será obrigatória a frequência às atividades correspondentes a cada disciplina, no mínimo a 75% das mesmas (Frequência Suficiente - FS), ficando nela reprovado o aluno que não comparecer a mais de 25% das atividades (Frequência Insuficiente - FI).

Serão realizadas duas provas e um trabalho:

- Prova Individual (P1) – 20%
- Prova Individual (P2) – 40%
- Trabalhos Complementares (TC) – 20%
- Trabalho (T1) – 20%

A nota mínima para aprovação na disciplina será  $MF \geq 6,0$  (seis) e Frequência Suficiente (FS). (Art. 69 e 72 da Res. nº 17/CUn/1997).

O aluno com Frequência Suficiente (FS) e média das notas de avaliações do semestre MF entre 3,0 e 5,5 terá direito a uma nova avaliação no final do semestre (REC), exceto as atividades constantes no art.70, § 2º. A Nota Final (NF) será calculada por meio da média aritmética entre a média das notas das avaliações parciais (MF) e a nota obtida na nova avaliação (REC). (Art. 70 e 71 da Res. nº 17/CUn/1997).

Ao aluno que não comparecer às avaliações ou não apresentar trabalhos no prazo estabelecido será atribuída nota 0 (zero). (Art. 70, § 4º da Res. nº 17/CUn/1997).

Observações:

Não há avaliação de recuperação nas disciplinas de caráter prático que envolve atividades de laboratório (Res.17/CUn/97).

Nova avaliação

O aluno, que por motivo de força maior e plenamente justificado, deixar de realizar atividades avaliativas previstas no plano de ensino, deverá formalizar pedido à Chefia do Departamento de Ensino ao qual a disciplina pertence, dentro do prazo de 3 (três) dias úteis, apresentando documentação comprobatória.

Horário de Atendimento:

Quartas-feiras: 16:00h às 18:00h – Sala 205 (CED/Bloco C)

## 6. CRONOGRAMA

Aula	Data	Conteúdo/Avaliações
1		Apresentação do plano de ensino Fundamentos da Programação – Algoritmos, Histórico sobre Linguagens de Programação
2		Estrutura de Sequenciação, Exercícios Estruturas de Decisão (if-else) - conceitos e exercícios
3		Processo de edição, compilação e execução - Python, Conceitos Básicos de Programação Imperativa (uso de variáveis, tipos de dados, comandos de atribuição, operadores aritméticos e lógicos)
4		Exercícios (atividade híbrida)
5		Estrutura de Repetição (For) – conceitos e exercícios de exemplo Estruturas de Repetição (While) – conceitos e exercícios de exemplo Exercícios (for, while)
6		Uso de Funções (função, parâmetro e argumento) Exercícios (Revisão Prova 1)
7		Prova 1
8		Coleções – Cadeia de Caracteres (String) Exercícios
9		Coleções Unidimensionais (Lista, Tupla) Lista de Exercícios
10		Set (conjuntos em Python) Coleções Bidimensionais (Matriz, Lista Composta) Exercícios (atividade híbrida)
11		Coleções Indexadas (Dicionário) Lista de exercícios
12		Prova 2
13		Conceitos Básicos de Orientação a Objetos (classe, objetos, atributos, métodos, argumentos e parâmetros), Modelo de Objetos (comunicação por troca de mensagens)
14		Apresentação do enunciado do Trabalho Prático, Como implementar uma Classe em Python
15		Herança Lista de Objetos (inserir, alterar, consultar, excluir) - (atividade híbrida)
16		Modelo de Objetos (hierarquia de agregação/decomposição, especialização/generalização), Polimorfismo
17		Apresentação do trabalho – (atividade híbrida) (T1)
18		Divulgação e discussão da avaliação final