

		<p align="center"> <b>Universidade Federal de Santa Catarina</b>  <b>Campus Araranguá - ARA</b>  <b>Centro de Ciências, Tecnologias e Saúde</b>  <b>Coordenadoria Especial Interdisciplinar em Tecnologias da Informação e Comunicação</b>  <b>Plano de Ensino</b> </p>	
<b>SEMESTRE 2020.2</b>			
<b>I. IDENTIFICAÇÃO DA DISCIPLINA</b>			
<b>CÓDIGO</b>	<b>NOME DA DISCIPLINA</b>	<b>HORAS-AULA - TEÓRICAS</b>	<b>HORAS-AULA - PRÁTICAS</b>
CIT7580	Algoritmos e Programação	2	4
<b>TOTAL DE HORAS-AULA SEMESTRAIS</b>	<b>HORÁRIO TURMAS TEÓRICAS</b>	<b>HORÁRIO TURMAS PRÁTICAS</b>	<b>MODALIDADE</b>
108	4as feiras- 18h30 – 20h10	4as feiras- 18h30 – 20h10	Ensino Remoto
<b>II. PROFESSOR(ES) MINISTRANTE(ES)</b>			
Cristian Cechinel			
<b>III. PRÉ-REQUISITO(S)</b>			
Esta disciplina não possui pré-requisitos			
<b>IV. CURSO(S) PARA O(S) QUAL(IS) A DISCIPLINA É OFERECIDA</b>			
TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO (noturno) [Campus Araranguá]			
<b>V. JUSTIFICATIVA</b>			
Esta disciplina de caráter técnico prepara o discente para o desenvolvimento de soluções computacionais usando técnicas de programação. Além dos aspectos fundamentais da lógica de programação, esta disciplina também foca o uso de uma primeira linguagem de programação.			
<b>VI. EMENTA</b>			
Conceito e estrutura de algoritmo. Pseudocódigo e fluxograma. Paradigma de programação estruturado x orientado a objetos. Linguagem Compilada x Interpretada. Noções de lógica de programação. Dados,			
<b>VII. OBJETIVOS</b>			
<p>Objetivo Geral:</p> <p>Tornar os alunos capazes de visualizar soluções computacionais para problemas através da aplicação dos conceitos da lógica de programação e dotá-los da capacidade de construção de programas, em linguagem de alto nível estruturada, que implementem as soluções vislumbradas.</p> <p>Objetivos Específicos:</p> <ul style="list-style-type: none"> <li>- Desenvolver o raciocínio lógico e abstrato do aluno;</li> <li>- Familiarizar o aluno com o modelo sequencial de computação;</li> <li>- Apresentar técnicas e linguagens para representação e construção de algoritmos simples;</li> <li>- Apresentar conceitos básicos de linguagens de programação;</li> <li>- Treinar o aluno no processo básico de desenvolvimento de software concepção, edição, execução e teste de programas de computador);</li> <li>- Capacitar o aluno no uso de uma linguagem de alto nível.</li> </ul>			
<b>VIII. CONTEÚDO PROGRAMÁTICO</b>			

Conteúdo Teórico seguido de Conteúdo Prático com desenvolvimento de problemas em computador:

**UNIDADE1: Introdução [16 horas-aula]**

- Conceito
- Algoritmo
- Funcionalidade de um algoritmo
- Estrutura de um algoritmo
- Pseudocódigo
- Introdução a algoritmos em alto nível
- Classificação das linguagens de programação com relação à similaridade com a linguagem natural
  - linguagem de máquina.
  - linguagem simbólica
  - linguagem de alto nível
- Exemplos de algoritmos

**UNIDADE 2: Conceituação de elementos básicos para construção de um algoritmo [16 horas-aula]**

- Constante
- Variável
- Identificador
- Palavra reservada
- Operadores aritméticos, de atribuição, relacionais e lógicos
- Parâmetros
- Tipos de dados primitivos
  - Lógico
  - Caractere
  - Inteiro
  - Real
- Conceito de lógica
- Método para construção de um algoritmo
- Estrutura de um pseudocódigo
- Construção de algoritmos em pseudocódigo
- Estrutura de E/S de dados – teclado e monitor

**UNIDADE 3: Estruturas de controle de fluxo: seleção [16 horas-aula]**

- Estruturas de seleção
  - Seleção simples
  - Seleção composta

**UNIDADE 4: Estruturas de controle de fluxo: repetição [16 horas-aula]**

- Estruturas de repetição
  - enquanto faça
  - para faça

**UNIDADE 5: Variáveis compostas [12 horas-aula ]**

- Vetores unidimensionais e multidimensionais
  - caracterização, declaração e indexação

**UNIDADE 6: Conceitos básicos de Linguagens de Programação [16 horas-aula]**

- Conceituação de Linguagem de Programação
- Atividades de programação com uma linguagem de programação
- Codificação, compilação/interpretação e execução

**UNIDADE 7: Modularização [16 horas-aula]**

- Modularização
  - Definição de módulos
  - Procedimentos/Funções/Métodos
  - Parâmetros (por valor e referência)

**IX. COMPETÊNCIAS/HABILIDADES**

**X. METODOLOGIA DE ENSINO / DESENVOLVIMENTO DO PROGRAMA**

A disciplina acontecerá no formato de sala de aula invertida em que o acadêmico deverá estudar o material disponível no ambiente Moodle antes das aulas síncronas que irão acontecer. Cabe ressaltar:

1. A disciplina acontecerá integralmente no Moodle.
2. Serão disponibilizados materiais de apoio e vídeo aulas sobre todos os conteúdos das disciplinas, além de exercícios práticos para serem respondidos pelos acadêmicos
3. Semanalmente também acontecerá uma aula síncrona em horário definido e sobre o conteúdo já estudado previamente pelos acadêmicos (sobretudo visualização de vídeo aulas).
4. Existem exercícios opcionais disponibilizados (desafios de programação) e uma lista de atividades online obrigatórias que serão contabilizadas na média final do acadêmico.

Requisitos de infraestrutura necessários para ministrar as aulas:

1. Acesso à Internet;
2. Ambiente Virtual de Aprendizagem - Moodle.

#### XI. METODOLOGIA E INSTRUMENTOS DE AVALIAÇÃO

A verificação do rendimento escolar será realizada por meio de atividades e trabalhos práticos disponibilizados no Moodle. Serão realizadas 11 avaliações, sendo divididas entre Atividades Online (8 atividades) e Trabalhos Práticos (3 trabalhos):

- AO1: Atividade Online 1
- AO2: Atividade Online 2
- AO3: Atividade Online 3
- AO4: Atividade Online 4
- AO5: Atividade Online 5
- AO6: Atividade Online 6
- AO7: Atividade Online 7
- AO8: Atividade Online 8
- TP1: Trabalho Prático 1
- TP2: Trabalho Prático 2
- TP3: Trabalho Prático 3

A Média Final (MF) será calculada da seguinte forma:

$$MF = (AO1 * 0.04 + AO2 * 0.04 + AO3 * 0.04 + AO4 * 0.04 + AO5 * 0.04 + AO6 * 0.04 + AO7 * 0.04 + AO8 * 0.04 + TP1 * 0.20 + TP2 * 0.20 + TP3 * 0.28)$$

A nota mínima para aprovação na disciplina será  $MF \geq 6,0$  (seis). Serão considerados frequentes os acadêmicos que entregarem 75% ou mais das atividades online propostas (AO) (de um total de 8 atividades os acadêmicos devem realizar um mínimo de 6). (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:

Avaliação de recuperação

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 fazer o pedido à Chefia do Departamento que oferece a disciplina, dentro do prazo de 3 (três) dias úteis, apresentando documentação comprobatória. O pedido de Nova Avaliação deve ser formalizado na Secretaria Integrada de Departamentos (SID).

Horários de atendimento:

#### XII. CRONOGRAMA

SEMANA	DATAS	ASSUNTO
1	01/02/2021 a 06/02/2021	Apresentação do plano de ensino e da disciplina Unidade 1: Introdução a algoritmos. Estrutura de um algoritmo. Pseudocódigo. Introdução a algoritmos em
		Unidade 1: Introdução a algoritmos. Estrutura de um

3	15/02/2021 a 20/02/2021	15/02/2021 a 20/02/2021 Unidade 6: Conceitos básicos de Linguagens de Programação
4	22/02/2021 a 27/02/2021	Parâmetros. Tipos de dados primitivos. Método para construção de um algoritmo Estrutura de um pseudocódigo. Construção de
5	01/03/2021 a 06/03/2021	Estrutura de seleção simples, composta e encadeada e exercícios. Unidade 4: Controle de Fluxo: repetição
6	08/03/2021 a 13/03/2021	Estruturas de repetição (enquanto-faça) Trabalho Prático 1 (TP1)
7	15/03/2021 a 20/03/2021	Estruturas de repetição (para-até) Atividade Online 4 (AO4)
8	22/03/2021 a 27/03/2021	Estruturas de repetição (para-até)
9	29/03/2021 a 03/04/2021	Unidade 5: Variáveis compostas Vetores unidimensionais – Listas Vetores unidimensionais – Listas
10	05/04/2021 a 10/04/2021	Trabalho Prático 2 (TP1)
11	12/04/2021 a 17/04/2021	Vetores unidimensionais – Listas Vetores unidimensionais – Listas. Matrizes
12	19/04/2021 a 24/04/2021	Vetores unidimensionais – Listas Vetores unidimensionais – Listas. Matrizes Atividade Online 6 (AO6)
13	26/04/2021 a 01/05/2021	Matrizes Atividade Online 7 (AO7)
14	03/05/2021 a 08/05/2021	Unidade 7: Modularização. Definição de módulos. Procedimentos/Funções/Métodos.
15	10/05/2021 a 15/05/2021	Procedimentos/Funções/Métodos. Parâmetros. Atividade Online 8 (AO8) Trabalho Prático 3 (TP3)
16	17/05/2021 a 22/05/2021	Avaliação de Recuperação - 4 a feira . Publicação das Notas
<b>Obs:</b> O calendário está sujeito a pequenos ajustes de acordo com as necessidades das atividades		
<b>XIII. FERIADOS PREVISTOS PARA O SEMESTRE</b>		
15/02/2021	Ponto facultativo Carnaval	
16/02/2021	Carnaval	
02/04/2021	Sexta-feira Santa	
03/04/2021	Aniversário de Araranguá	
21/04/2021	Tiradentes	
01/05/2021	Dia do Trabalho	
04/05/2021	Dia da Padroeira de Araranguá	
03/06/2021	Corpus Christi	
<b>XIV. BIBLIOGRAFIA BÁSICA</b>		
WAZLAWICK, Raul Sidnei. Introdução a Algoritmos e Programação com Python. Elsevier, 2017. 232 p. ISBN 9788535290516.		
MENEZES, Nilo Ney Coutinho. Introdução à programação com Python: algoritmos e lógica de programação para iniciantes. São Paulo: Novatec, 2010. 222 p. ISBN 9788575222508.		

FORBELLONE, André L. V.; EBERSPACHER, Henri F. Lógica de Programação. 3. ed. Pearson, 2005.

MCGUGAN, Will. Beginning Game Development with Python and Pygame: From Novice to Professional. Berkeley, CA: Apress, Inc., 2007. ISBN 9781430203254 Disponível em : <<http://dx.doi.org/10.1007/978-1-4302-0325-4>>. Acesso em : 9 out. 2009.

LOPES, Anita; GARCIA, Guto. Introdução à programação: 500 algoritmos resolvidos. Rio de Janeiro: Campus, 2002. 469 p. ISBN 8535210199.

**Professor(a):**

**Aprovado pelo Colegiado do Curso em \_\_\_/\_\_\_/\_\_\_ Presidente do Colegiado:**