



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS ARARANGUÁ
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE
CURSO DE TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO
PLANO DE ENSINO
2020.01

I. IDENTIFICAÇÃO DA DISCIPLINA:

CÓDIGO	NOME DA DISCIPLINA	Nº DE HORAS-AULA SEMANAIS REMOTAS	TOTAL DE HORAS-AULA SEMESTRAIS
CIT7580	Algoritmos e Programação	6	108

HORÁRIO		MODALIDADE
AULAS REMOTAS	4 ^{as} feiras- 18h30 – 20h10	Ensino Remoto

II. PROFESSOR(ES) MINISTRANTE(S)

Prof. Cristian Cechinel
Email: cristian.cechinel@ufsc.br

III. PRÉ-REQUISITO(S)

CÓDIGO	NOME DA DISCIPLINA
-	Esta disciplina não possui pré-requisitos

IV. CURSO(S) PARA O(S) QUAL(IS) A DISCIPLINA É OFERECIDA

Bacharelado em Tecnologias da Informação e Comunicação

V. JUSTIFICATIVA

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.

VI. EMENTA

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, expressões e algoritmos sequenciais. Comandos de entrada e saída, estruturas de controle de fluxo, operadores lógicos e aritméticos, estruturas de dados homogêneas e heterogêneas. Tipos definidos pelo usuário. Modularização. Introdução à uma Linguagem de Programação de alto nível. Atividades em laboratório com a linguagem selecionada.

VII. OBJETIVOS

Objetivo Geral:

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.

Objetivos Específicos:

- Desenvolver o raciocínio lógico e abstrato do aluno;
- Familiarizar o aluno com o modelo sequencial de computação;
- Apresentar técnicas e linguagens para representação e construção de algoritmos simples;
- Apresentar conceitos básicos de linguagens de programação;
- Treinar o aluno no processo básico de desenvolvimento de software concepção, edição, execução e teste de programas de computador);
- Capacitar o aluno no uso de uma linguagem de alto nível.

VIII. CONTEÚDO PROGRAMÁTICO

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

UNIDADE 1: 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. 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.

X. METODOLOGIA E INSTRUMENTOS DE AVALIAÇÃO

- A verificação do rendimento escolar será realizada por meio de atividades e trabalhos práticos disponibilizados no Moodle.
- Serão realizadas 10 avaliações, sendo divididas entre Atividades Online (8 atividades) e Trabalhos Práticos (2 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

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

$$MF = (AO1 * 0.05 + AO2 * 0.05 + AO3 * 0.05 + AO4 * 0.05 + AO5 * 0.05 + AO6 * 0.05 + AO7 * 0.05 + AO8 * 0.05 + TP1 * 0.25 + TP2 * 0.35)$$

- 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).
$$NF = \frac{(MF + REC)}{2}$$
- 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:

- Cristian Cechinel - Segunda-feira 18:00 – 20:00 Moodle

XI. CRONOGRAMA TEÓRICO/PRÁTICO		
AULA (semana)	DATA	ASSUNTO
1	05/03/2020 - 5ª feira	Apresentação do plano de ensino e da disciplina Unidade 1: Introdução a algoritmos. Estrutura de um algoritmo. Pseudocódigo. Introdução a algoritmos em alto nível.
2	11 e 12/03/2020 - 4ª e 5ª feira	Unidade 1: Introdução a algoritmos. Estrutura de um algoritmo. Pseudocódigo. Introdução a algoritmos em alto nível.
3	02/09/2020 - 4ª feira	Unidade 6: Conceitos básicos de Linguagens de Programação Conceituação de Linguagem de Programação. Classificação das linguagens de programação com relação à similaridade com a linguagem natural. Introdução ao ambiente de desenvolvimento IDLE – Codificação. Atividades de programação com uma linguagem de programação. Unidade 2: Visão geral das linguagens de programação. Declaração de variáveis. Palavras-reservadas. Operadores aritméticos, de atribuição, relacionais e lógicos Atividade Online 1 (AO1)
4	09/09/2020 - 4ª feira	Parâmetros. Tipos de dados primitivos. 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: Controle de Fluxo: seleção Estrutura de seleção simples, composta e encadeada e exercícios. Atividade Online 2 (AO2)
5	16/09/2020- 4ª feira	Estrutura de seleção simples, composta e encadeada e exercícios. Unidade 4: Controle de Fluxo: repetição Estruturas de repetição (enquanto-faça) Atividade Online 3 (AO3)
6	23/09/2020- 4ª feira	Estruturas de repetição (enquanto-faça)
7	30/09/2020- 4ª feira	Estruturas de repetição (para-até) Atividade Online 4 (AO4)
8	07/10/2020- 4ª feira	Estruturas de repetição (para-até)
9	14/10/2020- 4ª feira	Unidade 5: Variáveis compostas Vetores unidimensionais – Listas Vetores unidimensionais – Listas Atividade Online 5 (AO5)
10	21/10/2020- 4ª feira	Trabalho Prático 1 (TP1)
11	28/10/2020 (Feriado Dia do Servidor Público)	Vetores unidimensionais – Listas Vetores unidimensionais – Listas. Matrizes
12	04/11/2020- 4ª feira	Vetores unidimensionais – Listas Vetores unidimensionais – Listas. Matrizes Atividade Online 6 (AO6)
13	11/11/2020	Matrizes Atividade Online 7 (AO7)
14	18/11/2020	Unidade 7: Modularização. Definição de módulos. Procedimentos/Funções/Métodos.
15	25/11/2020	Procedimentos/Funções/Métodos. Parâmetros. Atividade Online 8 (AO8)
16	02/12/2020	Trabalho Prático 2 (TP2)
17	09/12/2020	Trabalho Prático 2 (TP2)
18	16/12/2020	Avaliação de Recuperação - 16.12.2018 - 4 a feira . Publicação das Notas

Obs: O calendário está sujeito a pequenos ajustes de acordo com as necessidades das atividades desenvolvidas.

XII. Feriados previstos para o semestre 2019.02

DATA	
07/09	Independência do Brasil
12/10	Nossa Senhora Aparecida
28/10	Dia do servidor público (Lei no 8.112 – art. 236)

02/11	Finados
15/11	Proclamação da República

XIII. BIBLIOGRAFIA BÁSICA

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.

MENEZES, Nilo Ney Coutinho. Introdução à programação com Python: algoritmos e lógica de programação para iniciantes. São Paulo: Novatec, 2010. 222 p. ISBN 9788575222508.

XIV. BIBLIOGRAFIA COMPLEMENTAR:

ARAÚJO, Everton Coimbra de. Algoritmos Fundamentos e Prática. Visual Books, 2007.

HETLAND, Magnus Lie. Beginning Python: From Novice to Professional. Second Edition. Berkeley, CA: Apress, 2008. ISBN9781430206347 Disponível em : <<http://dx.doi.org/10.1007/978-1-4302-0634-7>>. Acesso em : 9 out. 2009.

MEDINA, Marco; FERTIG, Cristina. Algoritmos e Programação – Teoria e Prática. 2. ed.. São Paulo: Novatec, 2006.

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).

XAVIER, Gley Fabiano Cardoso. Lógica de programação. 11. ed. São Paulo (SP): SENAC São Paulo, 2007.

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

Os livros acima citados constam na Biblioteca Universitária e Setorial de Araranguá. Algumas bibliografias também podem ser encontradas na Biblioteca Virtual da UFSC.

O referido programa de ensino foi aprovado na XX reunião ordinária do Colegiado do Departamento em xx de xxx de 2020.