



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE - CTS
DEPARTAMENTO DE COMPUTAÇÃO - DEC

PLANO DE ENSINO
SEMESTRE 2019.1

I. IDENTIFICAÇÃO DA DISCIPLINA:

CÓDIGO	NOME DA DISCIPLINA	Nº DE HORAS-AULASEMANAIS		TOTAL DE HORAS-AULA SEMESTRAIS
		TEÓRICAS	PRÁTICAS	
CIT7580	Algoritmos e Programação	0	6	108

HORÁRIO		MÓDULO
TURMAS TEÓRICAS	TURMAS PRÁTICAS	Semi-Presencial
	01652A – 3-2020-2 e 5-2020-2	2-0730-2

II. PROFESSOR(ES) MINISTRANTE(S)

Miriam Z. Parra Sejas
miriam.zareth.parra.sejas@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 Tecnologia 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. Pseudo-código 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. 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

UNIDADE 1 - Introdução [16 ha]

- Conceito
- Algoritmo
- Funcionalidade de um algoritmo
- Estrutura de algoritmo
- Pseudo-código
- 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 ha]

- 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 ha]

- Estruturas de seleção
 - Seleção simples
 - Seleção composta

UNIDADE 4 - Estruturas de controle de fluxo: repetição [16 ha]

- Estruturas de repetição
 - enquanto faça
 - para faça

UNIDADE 5 - Variáveis compostas [12 ha]

- Vetores unidimensionais e multidimensionais
 - caracterização, declaração e indexação

UNIDADE 6 - Conceitos básicos de Linguagens de Programação [16 ha]

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

UNIDADE 7 - Modularização [16 ha]

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

IX. METODOLOGIA DE ENSINO / DESENVOLVIMENTO DO PROGRAMA

1. Aulas expositivas intercaladas com discussões. Material de apoio postado no Moodle. Desenvolvimento de trabalhos e exercícios utilizando a ferramenta VisuAlg ou Scratch;
2. Atividades praticas no computador, utilizando o ambiente de desenvolvimento IDLE e a linguagem de Programação Python.

Requisitos de infraestrutura necessários para ministrar as aulas:

1. Datashow/projetor funcionando e com cabos HDMI/SVGA no comprimento adequado;
2. Acesso à Internet;
3. Laboratório de informática com computadores funcionando e em número adequado a quantidade de alunos;
4. Ambiente Virtual de Aprendizagem - Moodle.

X. METODOLOGIA E INSTRUMENTOS DE AVALIAÇÃO

- 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 três avaliações, sendo:
 - **AV1**: Avaliação Escrita 1 - referente a resolução de problemas utilizando algoritmos;
 - **AV2**: Avaliação Escrita 2 - referente a resolução de problemas utilizando algoritmos e a linguagem de programação Python;
 - **AV3**: Avaliação 3 - Trabalho Prático ou Avaliação Escrita, conforme necessidade de aprendizagem da turma, identificada pelo professor ao longo do semestre.

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

$$MF = [(AV1 + AV2) / 2] * 0,8 + AV3 * 0,2$$

- A nota mínima para aprovação na disciplina será **MF >= 6,0** (seis) e Frequência Suficiente (FS). (Art. 69 e 72 da Res. nº 17/CUn/1997)
- Caso seja encontrado **Cópia(s)** e/ou **Plágio(s)** em **qualquer avaliação**, seja em avaliação individual (AV1 e AV2) ou nos trabalhos e listas de exercícios (MT), o aluno estará automaticamente reprovado como nota ZERO (caso não esteja reprovado por FI)
- 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

- Pedidos de segunda avaliação somente para casos em que o aluno, por motivo de força maior e plenamente justificado, deixar de realizar avaliações previstas no plano de ensino; e deverá ser formalizado via requerimento de avaliação à Secretaria Acadêmica do Campus Araranguá dentro do prazo de 3 dias úteis apresentando comprovação. (Ver formulário)

Horários de atendimento aos alunos

- Segunda-feira 17:00 – 18:00. Jardim das Avenidas – Sala 307
- Terça-feira 17:00 – 18:00. Jardim das Avenidas – Sala 307

XI. CRONOGRAMA TEÓRICO/PRÁTICO

AULA (semana)	DATA	ASSUNTO
1	11/03 a 16/03	Apresentação do plano de ensino e da disciplina UNIDADE 1: Introdução a algoritmos Estrutura de um algoritmo Pseudo-código Introdução a algoritmos em alto nível Exercícios
2	18/03 a 23/03	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 Exercícios
3	25/03 a 30/03	SEATIC – Semana Acadêmica do Curso de TIC
4	01/04 a 06/04	UNIDADE 2: Visão geral das linguagens de programação Declaração de variáveis Palavra-reservada Operadores aritméticos, de atribuição, relacionais e lógicos Parâmetros
5	08/04 a 13/04	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 Exercícios
6	15/04 a 20/04	UNIDADE 3: Controle de Fluxo: seleção Estrutura de seleção simples, composta e encadeada Exemplos e exercícios Estrutura de seleção de múltipla escolha Exemplos e exercícios

7	22/04 a 27/04	UNIDADE 4: Controle de Fluxo: repetição Estrutura de repetição - Enquanto - faça Exemplos e exercícios
8	29/04 a 04/05	Estruturas de repetição - Para - faça Exemplos e exercícios
9	06/05 a 11/05	Estrutura de repetição (Exercícios) Revisão (Estrutura de repetição) Avaliação 1
10	13/05 a 18/05	UNIDADE 5: Variáveis compostas Vetores unidimensionais Exercícios
11	20/05 a 25/05	Listas - Pilhas e Filas Exercícios
12	27/05 a 01/06	Revisão e Exercícios Avaliação 2
13	03/06 a 08/06	UNIDADE 7: Modularização Definição de módulos Procedimentos/Funções/Métodos Exercícios
14	10/06 a 15/06	Parâmetros (por referência e valor) Exercícios
15	17/06 a 22/06	Revisão e Exercícios
16	24/06 a 29/06	Entrega dos Trabalhos (caso seja a opção) Avaliação 3 ou Apresentação dos Trabalhos
17	01/07 a 06/07	Prova substitutiva (Solicitação junto a secretária) Nova Avaliação (Prova de recuperação)
18	08/07 a 13/07	Publicação de Notas

Obs: O calendário está sujeito a pequenos ajustes de acordo com as necessidades das atividades desenvolvidas e em função da Semana Acadêmica do Curso de Tecnologias da Informação e Comunicação.

XII. Feriados previstos para o semestre 2019.1:

DATA	
03/04/2019	Aniversário Araranguá (Quarta)
19/04/2019	Sexta-feira Santa (Sexta)
20/04/2019	Dia não letivo (Sábado)
21/04/2019	Tiradentes / Páscoa (Domingo)
01/05/2019	Dia do Trabalhador (Quarta)
04/05/2019	Dia da Padroeira de Araranguá (Sábado)
20/06/2019	Corpus Christi (Quinta)
21/06/2019	Dia não letivo (Sexta)
22/06/2019	Dia não letivo (Sábado)

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 97888575222508.

XIV. BIBLIOGRAFIA COMPLEMENTAR:

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

HETLAND, Magñus 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.

Miriam Zareth Parra
Sejas:7043116
1127

Assinado de forma digital por Miriam Zareth Parra
Sejas:70431161127
Dados: 2019.02.14 18:05:52 -02'00'

Prof. Miriam Zareth Parra Sejas
Professor da Disciplina

/ / 2019

Aprovado pelo departamento em

/ / 2019

Aprovado pelo colegiado do curso de graduação em

26/02/2019
Prof. Dr. Wilson Gruber
Coordenador do Curso de
Tecnologias da Informação e Comunicação
Centro de Ciências, Tecnologias e Saúde
Port. nº /
SIAPE: 1926214 /CR