



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CAMPUS ARARANGUÁ - ARA

PLANO DE ENSINO

SEMESTRE 2016.1

I. IDENTIFICAÇÃO DA DISCIPLINA:

CÓDIGO	NOME DA DISCIPLINA	Nº DE HORAS-AULA SEMANAIS		TOTAL DE HORAS-AULA SEMESTRAIS
		TEÓRICAS	PRÁTICAS	
ARA7140	Programação em Computadores I	-	4	72

HORÁRIO		MODALIDADE
TURMAS TEÓRICAS	TURMAS PRÁTICAS	Presencial
	01652C 3.2020-2 e 5.1830-2	

II. PROFESSOR(ES) MINISTRANTE(S)

Prof. Jailson Torquato  
E-mail: jailson.torquato

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 de algoritmo. Pseudo-código e fluxograma. Estrutura de um algoritmo. identificadores, palavras reservadas, variáveis, constantes, declaração de variáveis, comandos de entrada e saída, estruturas de controle de fluxo, estruturas de dados homogêneas (vetores e matrizes) e heterogêneas (registros).

Tipos definidos pelo usuário. Modularização. Introdução à uma Linguagem de Programação de alto nível estruturada. 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 [8 horas-aula]**

- Conceito
- Algoritmo
- Funcionalidade de um algoritmo
- Exemplos de algoritmos

### **UNIDADE 2: Conceitos básicos de Linguagens de Programação [4 horas-aula]**

- Conceituação de Linguagem de Programação
- Classificação das linguagens de programação com relação à similaridade com a linguagem natural
  - linguagem de máquina
  - linguagem simbólica
  - linguagem de alto nível
- Atividades de programação com uma linguagem de programação
- Codificação, compilação/interpretação e execução

### **UNIDADE 3: Conceituação de elementos básicos para construção de um algoritmo [8 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 4: Estruturas de controle de fluxo: seleção [16 horas-aula]**

- Estruturas de seleção
  - Seleção simples
  - Seleção composta

### **UNIDADE 5: Estruturas de controle de fluxo: repetição [16 horas-aula]**

- Estruturas de repetição
  - enquanto faça
  - para faça

### **UNIDADE 6: Variáveis compostas [12 horas-aula]**

- Vetores unidimensionais e multidimensionais
  - caracterização, declaração e indexação
- Listas e Strings
  - caracterização, declaração e indexação

### **UNIDADE 7: Modularização [8 horas-aula]**

- 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;
2. Atividades práticas 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 e a linguagem de programação Python.  
**AV2:** Avaliação Escrita 2 - referente a resolução de problemas utilizando algoritmos e a linguagem de programação Python.  
**MT:** Trabalho Prático e Lista de Exercícios

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

$$MF = [(AV1 + AV2) / 2] * 0,7 + MT * 0,3$$

- 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).
- A **apresentação** do trabalho prático é **obrigatória**. A avaliação do trabalho prático é feita individualmente, mesmo que o trabalho seja feito em grupo. Desta forma, caso o aluno não apresente o trabalho, a nota **MT** é igual a 0 (zero).
- 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).

##### Avaliação de segunda chamada:

- Para pedido de segunda avaliação somente em casos em que o aluno, por motivo de força maior e plenamente justificado, deixar de realizar avaliações previstas no plano de ensino, deverá formalizar pedido de avaliação à coordenação do curso dentro do prazo de 3 dias úteis apresentando comprovação.

##### Horário de atendimento ao aluno:

- Quarta-feira 16:00 – 17:00. Jardim das Avenidas - Sala C-01 - Incubadora - Centro Araranguá
- Quinta-feira 16:00 – 18:00. Jardim das Avenidas - Sala C-01 - Incubadora - Centro Araranguá

## XI. CRONOGRAMA TEÓRICO/PRÁTICO

AULA (semana)	DATA	ASSUNTO
1	14/03/16 a 18/03/16	Apresentação do plano de ensino e da disciplinas <b>Unidade 1:</b> Introdução à arquitetura de computadores: memória, processador, bits e bytes Introdução a algoritmos
2	21/03/16 a 25/03/16	Estrutura de um algoritmo Pseudo-código Introdução a algoritmos em alto nível Introdução ao ambiente de desenvolvimento IDLE Exemplos e exercícios

3	28/03/16 a 01/04/16	<b>Unidade 2: Conceitos básicos de Linguagens de Programação</b> Conceituação de Linguagem de Programação Classificação das linguagens de programação com relação à similaridade com a linguagem natural Codificação, compilação/interpretação e execução
4	04/04/16 a 08/04/16	<b>Unidade 3:</b> Visão geral das linguagens de programação Declaração de variáveis. Chamadas de Funções Tipos de variáveis. Operador de atribuição. Resolução de exercícios. Estrutura de E/S de dados Exemplos e exercícios
5	11/04/16 a 15/04/16	<b>Feriado (07/09)</b> Estrutura de E/S de dados Operadores: aritméticos, lógicos e relacionais. Exemplos e exercícios <b>Unidade 4: Controle de Fluxo: seleção</b> Introdução ao Turtle, características, funções Estrutura de seleção simples, composta e encadeada e exercícios.
6	18/04/16 a 22/04/16	<b>II Semana Acadêmica de Tecnologias da Informação e Comunicação</b>
7	25/04/16 a 29/04/16	<b>Unidade 4: Controle de Fluxo: seleção</b> Estrutura de seleção simples, composta e encadeada e exercícios. Estrutura de seleção de múltipla escolha e exercícios. Exemplos e exercícios
8	02/05/16 a 06/05/16	Estrutura de seleção simples, composta e encadeada e exercícios. Estrutura de seleção de múltipla escolha e exercícios.
9	09/05/16 a 13/05/16	Estrutura de seleção Disponibilização dos temas dos <b>trabalhos</b> <b>Revisão</b> <b>Avaliação I</b>
10	16/05/16 a 20/05/16	<b>Unidade 5: Controle de Fluxo: repetição</b> Estruturas de repetição Exercícios
11	23/05/16 a 27/05/16	Estruturas de repetição Exercícios
12	30/05/16 a 03/06/16	Estruturas de repetição Exercícios <b>Unidade 6: Vetores, Listas e Strings</b> Listas: Operações e métodos Listas: cadeia de caracteres (String) Exercícios
13	06/06/16 a 10/06/16	Exercícios
14	13/06/16 a 17/06/16	<b>Unidade 7: Modularização</b> Definição de módulos Procedimentos/Funções/Métodos Parâmetros (por referência e valor) Exercícios
15	20/06/16 a 24/06/16	<b>Entrega dos Trabalhos no AVA</b> <b>Apresentação dos Trabalhos</b>
16	27/06/16 a 01/07/16	<b>Avaliação II</b>
17	04/07/16 a 08/07/16	<b>Nova Avaliação (Prova de recuperação): Prova Prática compreendendo todo o conteúdo da disciplina</b>
18	11/04/2016 a 15/07/16	<b>Publicação de Notas</b>

DATA	
24/03/2016	Dia não letivo
25/03/2016	Sexta Feira Santa
26/06/2016	Dia não letivo

03/04/2016	Dia não letivo - Campus de Araranguá: Aniversário de Araranguá
21/04/2016	Tiradentes
22/04/2016	Dia não letivo
23/04/2016	Dia não letivo
01/05/2016	Dia do Trabalhador
04/05/2016	Campus de Araranguá: dia da Padroeira da Cidade
26/05/2016	Corpus Christi
27 e 28/06/2016	Dias não letivos

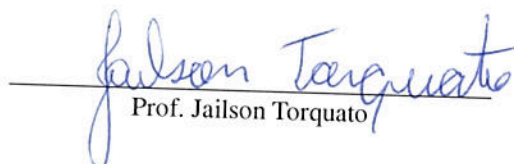
### XIII. BIBLIOGRAFIA BÁSICA

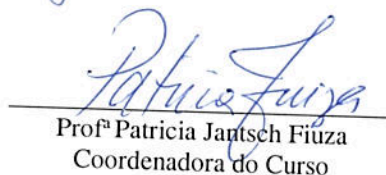
- HETLAND, Magnus Lie. **Beginning Python: From Novice to Professional**. Second Edition. Berkeley, CA: Apress, 2008. ISBN 9781430206347 Disponível em : <<http://dx.doi.org/10.1007/978-1-4302-0634-7>>. Acesso em : 9 out. 2009.
- FORBELLONE, André L. V.; EBERSPACHER, Henri F. **Lógica de Programação**. 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.

### XIV. BIBLIOGRAFIA COMPLEMENTAR:

- ARAÚJO, Everton Coimbra de. **Algoritmos Fundamentos e Prática**. Visual Books, 2007.
- 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 no acervo da disciplina, via sistema Moodle.

  
Prof. Jailson Torquato

  
Profª Patricia Jantsch Fiuza  
Coordenadora do Curso

**Profª. Patricia Jantsch Fiuza**  
Coordenadora do Curso de  
Tecnologias da Informação e Comunicação  
Portaria 101/2015/GR  
SIAPE: 2058903  
UFSC Centro Araranguá

Aprovado na Reunião do Colegiado do Curso \_\_\_/\_\_\_/\_\_\_