



PLANO DE CURSO

IDENTIFICAÇÃO

DISCIPLINA: Programação Imperativa **CÓDIGO:** COMP0334 **PERÍODO:** 2019.4
C.H.: 60 **Nº DE CRÉDITOS:** 4 **C.H. PRÁTICA:** 30 **PRÉ-REQUISITO(S):** -
TURMA: 04 **HORÁRIO:** 23456M2345
PROFESSORES: Alberto Costa Neto / Giovanny Fernando Lucero Palma / Kalil Araujo Bispo / Leonardo Nogueira Matos

EMENTA

Noções fundamentais sobre algoritmos e sobre a execução de programas. Análise e síntese de problemas. Identificadores, tipos, constantes, variáveis, tipos. Operadores e expressões. Comandos condicionais e de repetição. Variáveis compostas homogêneas e heterogêneas. Procedimentos, funções e passagem de parâmetros. Noções sobre o uso de arquivos em programação. Algoritmos básicos de ordenação. Recursividade. Uma linguagem imperativa. Convenções de código. Boas práticas de programação.

OBJETIVOS

- 1. Geral:** Apresentar os conceitos básicos e principais técnicas de desenvolvimento de programas de computador, tornando-o apto a compreendê-los e aplicá-los.
- 2. Específicos:**
 - Tornar o aluno capaz de implementar programas básicos usando uma linguagem de programação imperativa.
 - Habilitar o aluno a criar programas para executar computação científica na sua área de conhecimento.
 - Colocar em prática os conhecimentos aprendidos no curso, desenvolvendo aplicações de pequeno porte em Python.

CONTEÚDO PROGRAMADO (2H/AULA)

AULA 01:

- Introdução à disciplina (e-mail e site da disciplina, bibliografia, avaliação).
- O que é programar
- Motivação para programar

AULA 02: Hardware, software e princípios

AULA 03: Visão geral da linguagem Python

AULA 04: Preparação do ambiente de desenvolvimento

AULA 05:

- Constantes, variáveis, operador de atribuição e palavras reservadas
- Operador de atribuição
- Atribuição múltipla

AULA 06:



- Comandos de entrada e saída
- Comentários
- Tipos de dados, Conversão de tipos e operações com String

AULA 07: Expressões numéricas

AULA 08:

- Expressões relacionais e booleanas
- Ordem de avaliação e precedência de operadores

AULA 09:

- Comandos condicionais (if)
- Blocos de código (indentação)

AULA 10:

- Comandos condicionais aninhados e múltiplos (if / else / elif)
- Tratamento de exceções (try / except)

AULA 11:

- Comando de repetição while
- Comandos break e continue

AULA 12:

- Uso do laço While
- Funções predefinidas e Bibliotecas

AULA 13: Exercícios

AULA 14:

- Strings
- Comando de repetição for

AULA 15:

- Uso do Laço For
- Operadores e funções para Strings

AULA 16: Definição de funções, retorno de valores, argumentos e parâmetros

AULA 17: Funções recursivas

AULA 18:

- Listas
- Funções e Métodos de Listas

AULA 19:

- Aplicações de Listas
- Compreensão de Listas

AULA 20: Exercícios

AULA 21: Teste (1º Chamada)

AULA 22: Teste (2º Chamada)



AULA 23: Dicionários

AULA 24: Aplicações de Dicionários

AULA 25: Tuplas

AULA 26: Aplicações de Tuplas

AULA 27: Leitura e gravação de arquivos texto

AULA 28: Ordenação de dados

AULA 29: Exercícios

AULAS 30 a 33: Desenvolvimento de Projeto

AULAS 34 e 35: Apresentação do Projeto

AULA 36: Encerramento

METODOLOGIA

O conteúdo teórico estará disponível na Internet para que o aluno assista, possa rever e até adiantar o assunto, conforme sua disponibilidade. Além disso, o aluno terá acesso via Internet a um sistema que permite programar e, através da autoavaliação provida por ele, medir seu aprendizado. Com isso, o tempo de aula será focado em exercícios orientados pelos professores. Problemas mais complexos serão abordados no final da disciplina, visando preparar o aluno para implementar soluções no computador para problemas que encontrará na atividade profissional. O aprendizado dos alunos será avaliado através de teste e de um projeto realizado ao final da disciplina.

FORMA DE AVALIAÇÃO

A avaliação será feita através de teste e projeto, obedecendo à fórmula: $Nota\ Final = (NT + PROJ) / 2$.

Onde: NT = Nota do Teste e $PROJ$ = Nota do Projeto

Observações: Tanto o teste como o projeto terão uma etapa de submissão (online) e outra de arguição. Caso uma destas etapas não seja cumprida no prazo previsto, a nota correspondente será anulada. Haverá uma 2ª chamada para os alunos que, durante a realização do teste, enfrentem algum problema técnico, como perda de conexão ou defeito no computador.

Requisitos: O aluno precisa de um computador (*Notebook* ou *Desktop*) conectado à Internet. O computador deve possuir câmera ou o aluno deve dispor de um *smartphone* com câmera para que seja possível acompanhá-lo durante a prova e arguições. Caso não seja possível acompanhá-lo por vídeo durante a prova ou arguição, a nota será anulada.

RECURSOS DIDÁTICOS

As aulas serão ministradas por meio da Internet, utilizando um software de videoconferência, com horários agendados previamente. As ferramentas utilizadas serão:

- Ferramentas de Videoconferência: Zoom Cloud Meetings / Google Meet
- Editores de programas: Notepad++ ou Sublime Text.
- Interpretador da linguagem Python, que permite a verificação de erros de sintaxe e execução de programas em Python.
- *Apps* que permitem elaborar, executar e testar programas em *smartphones* e *tablets*.
- Ambientes Virtuais de Aprendizagem (AVA) SIGAA e Moodle
- Ambiente Virtual de Aprendizagem Móvel Moodley



- Questionários e Atividades via SIGAA
- Questionários com problemas de programação no site <http://thehuxley.com>

BIBLIOGRAFIA

1. Básica:

- Fundamentos da Programação de Computadores. Ana Fernanda Gomes Ascencio / Edilene Aparecida Veneruchi De Campos. 3ª edição; 2012, Pearson; ISBN 978-8564574168
- Algoritmos e Lógica de Programação. Marco A. Furlan de Souza, Marcelo M. Gomes, Marcio V. Soares, Ricardo Concilio. Editora Cengage Learning, 2ª edição, 2011.
- Algoritmos: Lógica para Desenvolvimento de Programação de Computadores. José Augusto N. G. Manzano, Jayr Figueiredo de Oliveira. Editora Érica, 17ª edição, 2005.
- Python for Everybody: Exploring Data Using Python 3. Charles R. Severance. CreateSpace Independent Publishing Platform; 1st. ed., 2016; ISBN: 978-1530051120

2. Complementar:

- Python for Informatics: Exploring Information. Charles R. Severance. CreateSpace Independent Publishing Platform; 1st. ed., 2013; ISBN: 978-1492339243
- Como pensar como um Cientista da Computação usando Python (traduzido). Allen Downey, Jeffrey Elkner, and Chris Meyers. 2002.
- Python para Desenvolvedores. Luiz Eduardo Borges. Rio de Janeiro; 2010; 978-85-909451-1-6.
- Introdução à Programação com Python. Nilo Ney Coutinho, 2ª edição, 2014, ISBN: 978-85-7522-408-3.
- Learning to Program Using Python. Cody Jackson. CreateSpace Independent Publishing Platform; ISBN: 9781461182054

Cidade Universitária “Prof. José Aloísio de Campos”, 08 de junho de 2020.

Alberto Costa Neto
Professor da Disciplina

Giovanny Fernando Lucero Palma
Professor da Disciplina

Kalil Araujo Bispo
Professor da Disciplina

Leonardo Nogueira Matos
Professor da Disciplina