



PLANO DE CURSO

IDENTIFICAÇÃO

DISCIPLINA: Programação Imperativa **CÓDIGO:** COMP0334 **PERÍODO:** 2020.1
C.H.: 60 **Nº DE CRÉDITOS:** 4 **C.H. PRÁTICA:** 30 **PRÉ-REQUISITO(S):** -
TURMAS: 03, 08 **HORÁRIO:** 6T1234 (03), 37M56 (08)
PROFESSOR: Alberto Costa Neto

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)

AULAS 01 e 02 (Síncrona: T08 20/10/2020 às 11:00 | T03 23/10/2020 às 15:00)

- Introdução à disciplina (e-mail e site da disciplina, bibliografia, avaliação).
- O que é programar
- Motivação para programar
- Hardware, software e princípios
- Visão geral da linguagem Python
- Preparação do ambiente de desenvolvimento

AULAS 03 e 04 (Síncrona: T08 27/10/2020 às 11:00 | T03 30/10/2020 às 15:00)

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



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

AULAS 05 e 06 (Síncrona: T08 03/11/2020 às 11:00 | T03 06/11/2020 às 15:00)

- Expressões numéricas
- Expressões relacionais e booleanas
- Ordem de avaliação e precedência de operadores
- Comandos condicionais (if)
- Blocos de código (indentação)
- Comandos condicionais aninhados e múltiplos (if / else / elif)
- Tratamento de exceções (try / except)

AULAS 07 e 08 (Síncrona: T08 10/11/2020 às 11:00 | T03 13/11/2020 às 15:00)

- Comando de repetição while
- Comandos break e continue
- Uso do laço While
- Funções predefinidas e Bibliotecas

AULAS 09 e 10 (Síncrona: T08 17/11/2020 às 11:00 | T03 20/11/2020 às 15:00)

- Strings
- Comando de repetição for
- Uso do Laço For
- Operadores e funções para Strings

AULAS 11 e 12 (Síncrona: T08 24/11/2020 às 11:00 | T03 27/11/2020 às 15:00)

- Definição de funções, retorno de valores, argumentos e parâmetros
- Funções recursivas

AULAS 13 e 14 (Síncrona: T08 01/12/2020 às 11:00 | T03 04/12/2020 às 15:00)

- Listas
- Funções e Métodos de Listas
- Aplicações de Listas
- Compreensão de Listas

AULAS 15 e 16 (Síncrona: T08 08/12/2020 às 11:00 | T03 11/12/2020 às 15:00)

- Dicionários
- Aplicações de Dicionários

AULAS 17 e 18 (Síncrona: T08 15/12/2020 às 11:00 | T03 18/12/2020 às 15:00)

- Tuplas
- Aplicações de Tuplas
- Leitura e gravação de arquivos texto
- Ordenação de dados



AULAS 19 e 20 (Síncrona: T08 19/01/2021 às 11:00 | T03 22/01/2021 às 15:00)

- Formação de Grupos para projeto
- Especificação do Projeto
- Desenvolvimento do Projeto
- Entrega das Videoaulas

AULAS 21 a 27 (Síncrona: T08 26/01/2021 às 11:00 | T03 29/01/2021 às 15:00)

- Desenvolvimento do Projeto
- Teste

AULAS 28 e 29 (Síncrona: T08 02/02/2021 às 11:00 | T03 05/02/2021 às 15:00)

- Apresentação do Projeto

AULA 30 (Síncrona: T08 09/02/2021 às 11:00 | T03 12/02/2021 às 15:00)

- 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 pelo professor. 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 atividades, teste e de um projeto realizado ao final da disciplina.

FORMA DE AVALIAÇÃO

A avaliação será feita através de atividades, teste e projeto, obedecendo à fórmula: **Nota Final = (NOTA1 + NOTA2) / 2**.

Onde:

NOTA1 = Nota das atividades: **SIGAA** (peso 1) + **The Huxley** (peso 3) e **Atividade livre** (peso 6)

NOTA2 = Nota do Projeto

SIGAA = Questionários semanais de múltipla escolha.

The Huxley = Questionários semanais com problemas de programação.

Atividade Livre = O aluno poderá escolher ser avaliado através de uma das formas abaixo:

- Teste síncrono que se dará através da resolução de um problema de programação ao vivo, preferencialmente em dupla (escolhida pelo professor).
- Apresentação de soluções para problemas selecionados, durante as aulas síncronas:
 - Para que esta atividade possa substituir o teste, o aluno deverá resolver ao menos 4 problemas em datas diferentes. Seu desempenho será avaliado tanto em relação à solução apresentada, como pelo conhecimento demonstrado sobre o assunto.
 - Caso o aluno resolva mais de 4 problemas, serão consideradas as 4 melhores notas obtidas. Caso tenha menos de 4 participações, o cálculo da nota continuará sendo sobre 4 problemas.



- Para garantir o máximo de participação da turma, terão prioridade aqueles alunos com menos participações na resolução dos problemas até o momento. Idealmente, cada aluno só poderá resolver um problema por aula. Porém, caso não haja outros interessados, um mesmo aluno poderá resolver mais de um problema, sendo computada uma das soluções e prevalecendo a melhor nota.
- (c) Preparação de uma videoaula e material de apoio (exemplos de código e texto explicativo) explicando um determinado assunto (selecionado pelo professor).

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 apenas 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: Google Meet.
- Editores de programas: Repl.it, IDLE, Notepad++ ou Sublime Text.
- Interpretador da linguagem Python.
- *Apps* que permitem elaborar, executar e testar programas em *smartphones* e *tablets*.
- Ambientes Virtuais de Aprendizagem (AVA) SIGAA e Google Classroom.
- Ambiente Virtual de Aprendizagem Móvel Moodle.
- 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”, 19 de outubro de 2020.

Alberto Costa Neto

Professor da Disciplina