

DISCIPLINA: Programação Imperativa

CÓDIGO: COMP0334

PERÍODO: 2021.2

C.H.: 60

Nº DE CRÉDITOS: 4

C.H. PRÁTICA: 30

PRÉ-REQUISITO(S): -

TURMA: T04

HORÁRIO: 24T12

PROFESSOR: Kalil Araujo Bispo

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.

1. Geral: Apresentar os conceitos básicos e principais técnicas de desenvolvimento de programas de computador, tornando-o apto a compreendê-los e aplicá-los.

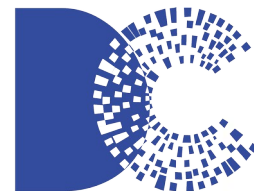
2. Específicos:

- 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.

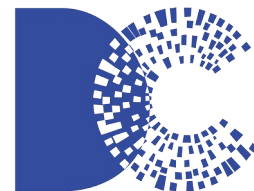
- Introdução à disciplina (e-mail e site da disciplina, bibliografia, avaliação).

- O que é programar
- Motivação para programar
- Hardware, software e princípios
- Visão geral da linguagem Python
- Preparação do ambiente de desenvolvimento

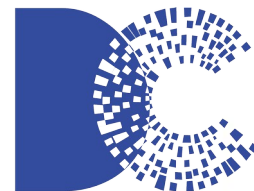
- Constantes, variáveis, operador de atribuição e palavras reservadas
- Operador de atribuição
- Atribuição múltipla
- Comandos de entrada e saída
- Comentários
- Tipos de dados, Conversão de tipos e operações com String
- Expressões numéricas



- Esclarecimento de dúvidas e resolução de exercícios
- Expressões relacionais e booleanas
- Ordem de avaliação e precedência de operadores
- Comandos condicionais (if)
- Blocos de código (indentação)
- Esclarecimento de dúvidas e resolução de exercícios
- Comandos condicionais aninhados e múltiplos (if / else / elif)
- Tratamento de exceções (try / except)
- Funções predefinidas e Bibliotecas
- Definição de funções, retorno de valores, argumentos e parâmetros
- Programação com abordagem Top-Down
- Comando de repetição while
- Comandos break e continue
- Uso do laço While
- Resolvendo problemas com Invariantes
- **Desafio de Programação 1**
- Strings
- Comando de repetição for
- Uso do Laço For
- Operadores e funções para Strings
- **Desafio de Programação 2**
- Esclarecimento de dúvidas e resolução de exercícios
- **1ª Prova**
- Listas
- Funções e Métodos de Listas



- Aplicações de Listas
- Compreensão de Listas
- Algoritmos de Ordenação (Inserção e Seleção)
- Esclarecimento de dúvidas e resolução de exercícios
- Algoritmo de Ordenação (Bolha)
- Busca Binária
- Matrizes
- Entrega das notas da 1ª Prova
- Resolução da 1ª Prova
- Funções recursivas
- Esclarecimento de dúvidas e resolução de exercícios
- Dicionários
- Aplicações de Dicionários
- Esclarecimento de dúvidas e resolução de exercícios
- **Desafio de Programação 3**
- Tuplas
- Aplicações de Tuplas
- Leitura e gravação de arquivos texto
- **Desafio de Programação 4**
- Esclarecimento de dúvidas e resolução de exercícios
- **2ª Prova**
- Esclarecimento de dúvidas e resolução de exercícios



- Esclarecimento de dúvidas e resolução de exercícios

- Entrega das notas da 2ª Prova
- Prova de Reposição

- Resolução da 2ª Prova
- Resolução da Prova de Reposição
- Entrega das notas da Prova de Reposição
- Encerramento

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 online será focado em resolver 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.

Haverá vários canais de comunicação entre docentes e discentes, dentre os quais podemos citar: (a) videoconferências e chats em horários pré-estabelecidas para retiradas de dúvidas e participação do aluno; (b) aplicativos de mensagens instantâneas, como Telegram; (c) troca de e-mails, Turma Virtual do SIGAA e/ou Classroom; (d) encontros presenciais previstos no plano de ensino.

O aprendizado dos alunos será avaliado através de provas presenciais durante a disciplina, que terão como foco avaliar a capacidade de criar algoritmos e de codificá-los em uma linguagem de programação.

Youtube, para exposição das videoaulas.

Computador, para desenvolvimento das aulas. Úteis à exposição de conteúdos organizados em slides, apresentação de exemplos ilustrativos e discussão das resoluções de exercícios.

Google Meet, para realização de videoconferências previstas no plano de ensino.

Google Jamboard, para apresentação dos objetos de ensino. Úteis à exposição de conteúdo, apresentação de exemplos ilustrativos e discussão das resoluções de exercícios.

Software de Apresentação, para apresentação dos objetos de ensino. Úteis à exposição de conteúdo, apresentação de exemplos ilustrativos e discussão das resoluções de exercícios.

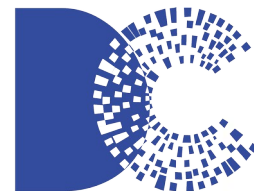
Editores de programas, para codificação dos programas, dentre os quais sugerimos utilizar o Repl.it, IDLE, Notepad++ ou Sublime Text.

Editores de para dispositivos móveis, como QPython e outros, para codificação utilizando celulares e tablets.

Interpretador da linguagem Python, para execução dos programas desenvolvidos.

Juiz online The Huxley, para realização de exercícios e desafios de programação, além de apoio nas provas.

Turma Virtual do SIGAA-UFS e/ou Classroom, com material de apoio (slides, exercícios, plano de ensino)



para download, avisos, e contatos docentes.

Fórum de dúvidas, como ambiente participativo e colaborativo entre os estudantes bem como para acompanhamento tutorial realizado pelo docente.

Conteúdos multimídia, Fóruns, Chats, Videoaulas.

A avaliação será feita através de testes presenciais, obedecendo à fórmula: **Nota Final** = $(NT1 + NT2) / 2$.

Onde: *NT1* = Nota do 1º Teste e *NT2* = Nota do 2º Teste

Observação: Haverá um teste de reposição no final do semestre apenas para os alunos com falta justificada em algum teste, desde que a justificativa esteja prevista nas normas acadêmicas.

Pontuação Extra: Durante o semestre haverá competições no estilo maratona de programação (desafios), no qual o aluno poderá obter até 1,5 (um ponto e meio) extras, ou seja, será uma nota adicionada à sua média geral, conforme sua classificação final na competição.

Regras da Pontuação Extra:

O requisito para concorrer à pontuação extra é ter no mínimo 60 pontos na primeira unidade e 100 pontos na segunda unidade no ranking *Top Coder* do site The Huxley, no dia da prova presencial da respectiva unidade. Aqueles que satisfizerem este requisito serão ordenados de forma decrescente em relação a esta pontuação e classificados em 4 grupos de tamanhos iguais, conforme a pontuação obtida. Caso não seja possível fazer uma divisão exata, o resto da divisão ficará no grupo 4.

Grupo 1: 1,5 pontos extras iniciais.

Grupo 2: 1,0 ponto extra inicial.

Grupo 3: 0,5 ponto extra inicial.

Grupo 4: sem pontos extras.

Além disso, o aluno terá que participar dos desafios de programação, cujas notas serão usadas para calcular a pontuação extra final do aluno, conforme fórmulas abaixo:

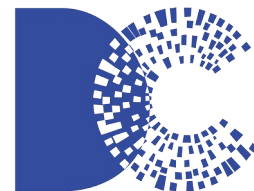
Pontuação extra unidade 1 = Pontuação extra inicial x (nota desafio 1 + nota desafio 2)/20

Pontuação extra unidade 2 = Pontuação extra inicial x (nota desafio 3 + nota desafio 4)/20

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”, 24 de janeiro de 2022.
