



PLANO DE CURSO

IDENTIFICAÇÃO

DISCIPLINA: Programação Imperativa **CÓDIGO:** COMP0334 **PERÍODO:** 2026.2
C.H.: 60 **Nº DE CRÉDITOS:** 4 **C.H. PRÁTICA:** 30 **PRÉ-REQUISITO(S):** -
TURMA: T08 e T09 **HORÁRIO:** 35M34(T08) e 35M56(T09) **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 (1H30/AULA)

AULA 01 (Presencial: 18/08/2026 - terça)

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

AULA 02 (Presencial: 20/08/2026 - quinta)

- Visão geral da linguagem Python
- Preparação do ambiente de desenvolvimento

AULA 03 (Presencial: 25/08/2026 - terça)

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



AULA 04 (Presencial: 27/08/2026 - quinta)

- Expressões relacionais e booleanas
- Ordem de avaliação e precedência de operadores
- Comandos condicionais (if)
- Blocos de código (indentação)
- Esclarecimento de dúvidas e resolução de exercício

AULA 05 (Presencial: 01/09/2026 - terça)

- Comandos condicionais aninhados e múltiplos (if / else / elif)

AULA 06 (Presencial: 03/09/2026 - quinta)

- Tratamento de exceções (try / except)
- **Prova 1**

SEM AULA (08/09/2026 – terça – feriado)

AULA 07 (Presencial: 10/09/2026 - quinta)

- Funções predefinidas e bibliotecas
- Definição de funções, retorno de valores, argumentos e parâmetros
- Esclarecimento de dúvidas e resolução de exercícios

AULA 08 (Presencial: 15/09/2026 - terça)

- Comando de repetição while
- Comandos break e continue
- Uso do laço while

AULA 09 (Presencial: 17/09/2026 - quinta)

- **Prova 2**
- Esclarecimento de dúvidas e resolução de exercícios

AULA 10 (Presencial: 22/09/2026 – terça)

- Strings
- Operadores e funções para Strings

AULA 11 (Presencial: 24/09/2026 - quinta)

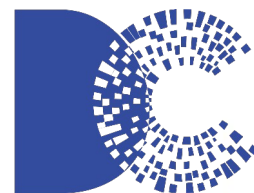
- Comando de repetição for
- Uso do Laço For

AULA 12 (Presencial: 29/09/2026 - terça)

- Listas
- Funções e Métodos de Listas

AULA 13 (Presencial: 01/10/2026 - quinta)

- Aplicações de Listas
- Esclarecimento de dúvidas e resolução de exercícios



AULA 14 (Presencial: 06/10/2026 - terça)

- Compreensão de Listas

AULA 15 (Presencial: 08/10/2026 – quinta)

- **Prova 3**
- Esclarecimento de dúvidas e resolução de exercícios

AULA 16 (Presencial: 13/10/2026 - terça)

- Conjuntos

AULA 17 (Presencial: 15/10/2026 - quinta)

- Algoritmos de Ordenação (Inserção, Seleção e Bolha)
-

AULA 18 (Presencial: 20/10/2026 - terça)

- Busca Binária

AULA 19 (Presencial: 22/10/2026 - quinta)

- **Prova 4**
- Esclarecimento de dúvidas e resolução de exercícios

AULA 20 (Presencial: 27/10/2026 - terça)

- Matrizes

AULA 21 (Presencial: 29/10/2026 - quinta)

- Esclarecimento de dúvidas e resolução de exercícios

AULA 22 (Presencial: 03/11/2026 - terça)

- Funções recursivas

AULA 23 (Presencial: 05/11/2026 - quinta)

- **Prova 5**
- Esclarecimento de dúvidas e resolução de exercícios

AULA 24 (Presencial: 10/11/2026 - terça)

- Dicionários
- Aplicações de Dicionários

AULA 25 (Presencial: 12/11/2026 - quinta)

- Esclarecimento de dúvidas e resolução de exercícios

AULA 26 (Presencial: 17/11/2026 - terça)

- Tuplas
- Aplicações de Tuplas

AULA 27 (Presencial: 19/11/2026 - quinta)

- Leitura e gravação de arquivos texto



AULA 28 (Presencial: 24/11/2026 - terça)

- **Prova 6**

AULA 29 (Presencial: 26/11/2026 - quinta)

- **Prova de Reposição**

AULA 30 (Presencial: 01/12/2026 - terça)

- Entrega das notas da Prova de Reposição
- Encerramento

HORA-TRABALHO

A hora-trabalho é de 15 min por aula, ou seja, 30 min por dia de aula, e envolverá atividades práticas de programação divididas em 15(quinze) questionários na plataforma The Huxley, leitura de capítulos de livros, além de vídeos relacionados.

METODOLOGIA

O conteúdo teórico está disponível no Youtube na forma de videoaula 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.

O tempo de aula será utilizado para avaliar os exercícios resolvidos pelos alunos e, de forma cooperativa, criar soluções para os problemas de programação apresentados com a orientação do 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 provas presenciais e exercícios, que terão como foco avaliar a capacidade de criar algoritmos e de codificá-los em uma linguagem de programação.

TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO

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.

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 VS Code, PyCharm, 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 pro-



vas.

Koude, para realização das provas em ambiente de laboratório.

Google Colab: serviço em nuvem gratuito baseado no Jupyter Notebook que permite executar código fonte em Python, com acesso gratuito online e pode ser intercalado com textos formatados. Serão utilizados para apresentar conceitos de forma gradativa e exercitando trechos de código fonte como exercícios aplicados.

Turma Virtual do SIGAA-UFS e/ou Classroom, com material de apoio (slides, exercícios, plano de ensino) para download, avisos, e contatos docentes.

RECURSOS DE APRENDIZAGEM

Conteúdos multimídia, Exercícios, Livros, Videoaulas.

FORMA DE AVALIAÇÃO

A avaliação será feita através de testes presenciais, obedecendo à fórmula: **Nota Final** = $(N1 + N2) / 2$.

Onde:

$N1$ = Média das provas da 1ª unidade

$N2$ = Média das provas da 2ª unidade

Em cada unidade, haverá pelo menos 3 provas obrigatórias e com datas definidas no plano de ensino. Pode haver provas surpresa, mas estas serão consideradas como extra e servem apenas para descartar notas mais baixas em provas obrigatórias ou repor falta em alguma prova obrigatória, ou seja, se o aluno tirar uma nota menor na prova surpresa do que todas as notas das provas obrigatórias, a nota da prova surpresa não será considerada no cálculo da média.

Observação: Haverá uma prova de reposição no final do semestre apenas para os alunos com falta justificada em alguma prova obrigatória, desde que a justificativa esteja prevista nas normas acadêmicas. Entretanto, caso o aluno tenha feito provas surpresa e atingido 3 provas por unidade, não terá direito à prova de reposição por já ter tido oportunidade de fazer a avaliação perdida em outro dia.

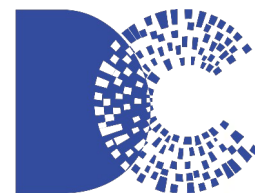
BIBLIOGRAFIA

1. Básica:

- Python Para Todos: Explorando Dados com Python 3. Charles R. Severance. Publicação independente; 1ª edição, 2020; ISBN: 979-8635191408
- Think Python: How to Think Like a Computer Scientist. Allen Downey. 2024. 3rd Edition. Disponível em <https://alldowney.github.io/ThinkPython/>

2. Complementar:

- 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.



Cidade Universitária “Prof. José Aloísio de Campos”, 17 de agosto de 2026.

Alberto Costa Neto