SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO – DComp

Departamento de Computação/UFS

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: T06 **HORÁRIO:** 2N12 4N34

PROFESSORES: Alberto Costa Neto, Victor Flavio, Yuri Passos

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: 17/08/2026 - segunda)

- 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: 19/08/2026 - quarta)

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

AULA 03 (Presencial: 24/08/2026 - segunda)

- 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: 26/08/2026 - quarta)

- 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)

AULA 05 (Presencial: 31/08/2026 - segunda)

- Funções predefinidas e bibliotecas
- Definição de funções, retorno de valores, argumentos e parâmetros

AULA 06 (Presencial: 02/09/2026 - quarta)

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

SEM AULA (07/09/2026 – segunda – feriado)

AULA 07 (Presencial: 09/09/2026 - quarta)

- ***Prova 1***

SEM AULA (14/09/2026 – segunda)

SEM AULA (16/09/2026 – quarta)

AULA 08 (Presencial: 21/09/2026 – segunda)

- Comando de repetição while
- Comandos break e continue

AULA 09 (Presencial: 23/09/2026 - quarta)

- Uso do laço while

AULA 10 (Presencial: 28/09/2026 - segunda)

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

AULA 11 (Presencial: 30/09/2026 - quarta)

- Esclarecimento de dúvidas e resolução de exercícios

AULA 12 (Presencial: 05/10/2026 – segunda)

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

AULA 13 (Presencial: 07/10/2026 – quarta)

- Conjuntos

SEM AULA (12/10/2026 – segunda – feriado)

AULA 14 (Presencial: 14/10/2026 - quarta)

- Algoritmos de Ordenação (Inserção, Seleção e Bolha)

AULA 15 (Presencial: 19/10/2026 - segunda)

- Busca Binária

AULA 16 (Presencial: 21/10/2026 - quarta)

- ***Prova 2***

AULA 17 (Presencial: 26/10/2026 - segunda)

- Matrizes

SEM AULA (28/10/2026 – quarta – recesso acadêmico)

SEM AULA (02/11/2026 – segunda – feriado)

AULA 18 (Presencial: 04/11/2026 - quarta)

- Matrizes

AULA 19 (Presencial: 09/11/2026 - segunda)

- Funções recursivas

AULA 20 (Presencial: 11/11/2026 - quarta)

- Dicionários
- Aplicações de Dicionários

AULA 21 (Presencial: 16/11/2026 - segunda)

- Tuplas
- Aplicações de Tuplas

AULA 22 (Presencial: 18/11/2026 - quarta)

- Leitura e gravação de arquivos texto

AULA 23 (Presencial: 23/11/2026 - segunda)

- ***SEMAC***

AULA 24 (Presencial: 25/11/2026 - quarta)

- ***SEMAC***

AULA 25 (Presencial: 30/11/2026 - segunda)

- Esclarecimento de dúvidas e resolução de exercícios

AULA 26 (Presencial: 02/12/2026 - quarta)

- ***Prova 3***

AULA 27 (Presencial: 07/12/2026 - segunda)

- Entrega das notas da Prova 3

AULA 28 (Presencial: 09/12/2026 - quarta)

- ***Prova de Reposição***

AULA 29 (Presencial: 14/12/2026 - segunda)

- Entrega das notas da Prova de Reposição

AULA 30 (Presencial: 16/12/2026 - segunda)

- 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 16(dezesseis) 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 ta-

blets.

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.

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+N3) / 3***.
Onde:

N1 = Nota da 1º prova

N2 = Nota da 2º prova

N3 = Nota da 3º prova

Em cada unidade, haverá uma prova com datas definidas no plano de ensino.

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.

BIBLIOGRAFIA

1. Básica:

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

2. Complementar:

- Fundamentos da Programação de Computadores. Ana Fernanda Gomes Ascencio / Edilene Aparecida Veneruchi De Campos. 3˚ 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”, 19 de agosto de 2026.

Alberto Costa Neto	**Victor Flavio Araujo**	**Yuri Passos**