



## PLANO DE CURSO

### IDENTIFICAÇÃO

**DISCIPLINA:** Programação Imperativa **CÓDIGO:** COMP0334 **PERÍODO:** 2025.2  
**C.H.:** 60 **Nº DE CRÉDITOS:** 4 **C.H. PRÁTICA:** 30 **PRÉ-REQUISITO(S):** -  
**TURMAS:** 09, 10 e 15 **HORÁRIO:** 35M12(T09), 35M34(T10) 35M56(T15)  
**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:** 07/10/2025 - 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:** 09/10/2025 - quinta)

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

#### AULA 03 (**Presencial:** 14/10/2025 - terça)

- 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 exercício

**AULA 04 (Presencial: 16/10/2025 - 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: 21/10/2025 - terça)**

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

**AULA 06 (Presencial: 23/10/2025 - quinta)**

- Esclarecimento de dúvidas e resolução de exercício

**AULA 07 (Presencial: 28/10/2025 - terça)**

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

**AULA 08 (Presencial: 30/10/2025 - quinta)**

- Comando de repetição while
- Comandos break e continue
- Uso do laço While
- Esclarecimento de dúvidas e resolução de exercícios

**AULA 09 (Presencial: 04/11/2025 - terça)**

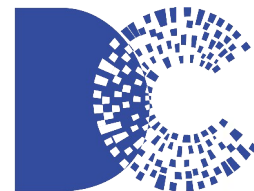
- Strings
- Operadores e funções para Strings
- Esclarecimento de dúvidas e resolução de exercícios

**AULA 10 (Presencial: 06/11/2025 - quinta)**

- Comando de repetição for
- Uso do Laço For
- Esclarecimento de dúvidas e resolução de exercícios

**AULA 11 (Presencial: 11/11/2025 - terça)**

- Esclarecimento de dúvidas e resolução de exercícios



**AULA 12 (Presencial: 13/11/2025 - quinta)**

- Esclarecimento de dúvidas e resolução de exercícios

**AULA 13 (Presencial: 18/11/2025 - terça)**

- 1ª Prova

**SEM AULA (20/11/2025 - quinta - feriado)**

**AULA 14 (Presencial: 25/11/2025 - terça)**

- SEMAC

**AULA 15 (Presencial: 27/11/2025 - quinta)**

- SEMAC

**AULA 16 (Presencial: 02/12/2025 - terça)**

- Listas
- Funções e Métodos de Listas
- Aplicações de Listas
- Compreensão de Listas
- Entrega das notas da 1ª Prova
- Resolução da 1ª Prova

**AULA 17 (Presencial: 04/12/2025 - quinta)**

- Conjuntos
- Esclarecimento de dúvidas e resolução de exercícios

**AULA 18 (Presencial: 09/12/2025 - terça)**

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

**AULA 19 (Presencial: 11/12/2025 - quinta)**

- Esclarecimento de dúvidas e resolução de exercícios

**AULA 20 (Presencial: 16/12/2025 - terça)**

- Matrizes

**AULA 21 (Presencial: 18/12/2025 - quinta)**

- Funções recursivas

**SEM AULA (23/12/2025 - terça - recesso natalino)**

**SEM AULA (23/12/2025 - quinta - recesso natalino)**

**SEM AULA (30/12/2025 - terça - recesso natalino)**

**SEM AULA (01/01/2026 - quinta - recesso natalino)**





**SEM AULA (06/01/2026 - terça – férias coletivas)**

**SEM AULA (08/01/2026 - quinta – férias coletivas)**

**AULA 22 (Presencial: 13/01/2026 - terça)**

- Dicionários
- Aplicações de Dicionários

**AULA 23 (Presencial: 15/01/2026 - quinta)**

- Esclarecimento de dúvidas e resolução de exercícios

**AULA 24 (Presencial: 20/01/2026 - terça)**

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

**AULA 25 (Presencial: 22/01/2026 - quinta)**

- Esclarecimento de dúvidas e resolução de exercícios

**AULA 26 (Presencial: 27/01/2026 - terça)**

- Esclarecimento de dúvidas e resolução de exercícios

**AULA 27 (Presencial: 29/01/2026 - quinta)**

- Esclarecimento de dúvidas e resolução de exercícios

**AULA 28 (Presencial: 03/02/2026 - terça)**

- 2ª Prova

**AULA 29 (Presencial: 05/02/2026 - quinta)**

- Esclarecimento de dúvidas e resolução de exercícios

**AULA 30 (Presencial: 10/02/2026 - terça)**

- Prova de Reposição

**AULA 31 (Presencial: 12/02/2025 - quinta)**

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

### 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:**



$N1$  = Nota do 1º Teste (peso 8) + Exercícios (peso 2)

$N2$  = Nota do 2º Teste (peso 8) + Exercícios (peso 2)

**Exercícios:** Haverá uma série de questionários no The Huxley, cada um com vários problemas de programação. A nota dos exercícios será calculada de acordo com a pontuação obtida em cada questionário, variando de 0 a 10. A nota dos exercícios da unidade será a média das notas dos questionários. Durante as aulas, o professor selecionará alunos para apresentarem suas soluções para os exercícios e responder questões sobre o assunto. Caso o aluno se negue a apresentar ou não o faça de forma que fique claro seu entendimento da solução e conhecimento sobre o assunto, o aluno perderá os pontos referentes aos Exercícios, passando a sua prova a ter peso 10. Presume-se que, se o aluno resolveu de fato o exercícios, o mesmo deve ser capaz de explicar seu raciocínio lógico e as construções de linguagem utilizadas. Caso o aluno apresente todas as vezes seguintes com sucesso, pode recuperar a contagem de pontos dos exercícios.

**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.

## 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 Para Todos: Explorando Dados com Python 3. Charles R. Severance. Publicação independente; 1ª edição, 2020; ISBN: 979-8635191408

### 2. Complementar:

- 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", 06 de outubro de 2025.

---

**Alberto Costa Neto**