



**UNIVERSIDADE FEDERAL DE SERGIPE**  
**PRÓ-REITORIA DE GRADUAÇÃO**  
**DEPARTAMENTO DE COMPUTAÇÃO**

**PLANO DE CURSO**

IDENTIFICAÇÃO			
<b>Disciplina</b>	Programação Imperativa	<b>Código/Turma</b>	COMP0334-T04/T08/T12
<b>Pré-requisito(s)</b>	—	<b>Carga horária</b>	60
<b>PEL</b>	4.0.0	<b>Créditos</b>	4
<b>Professor(es)</b>	Alberto Costa Neto Kalil Araújo Bispo	<b>Semestre</b>	2017.2
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			
<b>1. GERAL:</b> Apresentar os conceitos básicos e principais técnicas de desenvolvimento de programas de computador, tornando-o apto a compreendê-los e aplicá-los. <b>2. ESPECÍFICOS:</b> <ul style="list-style-type: none"><li>• Tornar o aluno capaz de implementar programas básicos usando uma linguagem de programação imperativa.</li><li>• Habilitar o aluno a criar programas para executar computação científica na sua área de conhecimento.</li><li>• Colocar em prática os conhecimentos aprendidos no curso, desenvolvendo aplicações de pequeno porte em Python.</li></ul>			
CONTEÚDO PROGRAMÁTICO			
<ol style="list-style-type: none"><li>1. Introdução à disciplina (e-mail e site da disciplina, bibliografia, avaliação).</li><li>2. O que é programar</li><li>3. Motivação para programar</li><li>4. Hardware, software e princípios</li><li>5. Visão geral da linguagem Python</li><li>6. Preparação do ambiente de desenvolvimento</li><li>7. Constantes, variáveis, operador de atribuição e palavras reservadas</li><li>8. Operador de atribuição</li><li>9. Atribuição múltipla</li><li>10. Comandos de entrada e saída</li><li>11. Comentários</li><li>12. Tipos de dados, Conversão de tipos e operações com String</li><li>13. Expressões numéricas</li><li>14. Expressões relacionais e booleanas</li><li>15. Ordem de avaliação e precedência de operadores</li><li>16. Comandos condicionais (if)</li><li>17. Blocos de código (indentação)</li><li>18. Comandos condicionais aninhados e múltiplos (if / else / elif)</li><li>19. Tratamento de exceções (try / except)</li><li>20. 1º Teste</li><li>21. Comando de repetição while</li><li>22. Comandos break e continue</li><li>23. Uso do laço While</li><li>24. Funções predefinidas e Bibliotecas</li><li>25. Strings</li><li>26. Comando de repetição for</li><li>27. Uso do Laço For</li><li>28. Operadores e funções para Strings</li></ol>			

29. Definição de funções, retorno de valores, argumentos e parâmetros
30. Funções recursivas
31. 2º Teste
32. Listas
33. Aplicações de Listas
34. Dicionários
35. Aplicações de Dicionários
36. Tuplas
37. Aplicações de Tuplas
38. Leitura e gravação de arquivos texto
39. 3º Teste
40. Teste de reposição

### 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 e monitor/tutor. 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 testes.

### RECURSOS DIDÁTICOS

As aulas serão ministradas em sala de aula e/ou laboratório (caso haja disponibilidade) com auxílio de data show, quadro e as ferramentas para programação de computadores, são elas:

- Editores de programas: Notepad++ ou Sublime Text.
- Interpretador da linguagem Python, que permite a verificação de erros de sintaxe e execução de programas em Python.
- *Apps* que permitem elaborar, executar e testar programas em *smartphones* e *tablets*.
- Ambientes Virtuais de Aprendizagem (AVA) SIGAA e Moodle
- Ambiente Virtual de Aprendizagem Móvel Moodley
- Questionários e Atividades via SIGAA
- Questionários com problemas de programação no site <http://thehuxley.com>

### HORÁRIOS DE ATENDIMENTO

Os horários de atendimento do professor e dos monitores serão divulgados e mantidos sempre atualizados pelo SIGAA.

### FORMA DE AVALIAÇÃO

A avaliação será feita através de testes, obedecendo à fórmula: **Nota Final** =  $(NT1 + NT2 + NT3) / 3$ .  
**Onde:** NT1 = Nota do 1º Teste, NT2 = Nota do 2º Teste e NT3 = Nota do 3º Teste

**Observação:** Haverá um teste de reposição para os alunos com falta justificada em algum teste, conforme previsto nas normas acadêmicas. Caso o aluno tenha feito todos os testes e obtido uma nota inferior a 5,0 em pelo menos um deles, poderá fazer o teste de reposição para tentar substituir a nota mais baixa. Como PI tem conteúdo inerentemente acumulativo, o teste de reposição englobará todo o assunto da disciplina.

### REFERÊNCIAS BIBLIOGRÁFICAS

#### 1. REFERÊNCIAS BÁSICAS:

- Python for Everybody: Exploring Data Using Python 3. Charles R. Severance. CreateSpace Independent Publishing Platform; 1st. ed., 2016; ISBN: 978-1530051120
- 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
- Learning to Program Using Python. Cody Jackson.
- CreateSpace Independent Publishing Platform; ISBN: 9781461182054

#### 2. REFERÊNCIAS COMPLEMENTARES:

Fundamentos da Programação de Computadores. Ana Fernanda Gomes Ascencio / Edilene Aparecida Veneruchi De Campos. 3º edição; 2012, Pearson; ISBN 978-8564574168