



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

**PLANO DE CURSO**

<b>IDENTIFICAÇÃO</b>			
<b>Disciplina</b>	Introdução à Ciência da Computação	<b>Código/Turma</b>	COMP0100-T01 COMP0208-T02
<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 Carlos Alberto Estombelo Montesco Kleber Tarcísio Oliveira Santos	<b>Semestre</b>	2016.1
<b>EMENTA</b>			
Conceitos gerais. Algoritmos e fluxogramas. Programação científica. Funções e procedimentos.			
<b>OBJETIVOS</b>			
<p><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.</p> <p><b>2. ESPECÍFICOS:</b></p> <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>			
<b>CONTEÚDO PROGRAMÁTICO</b>			
<ol style="list-style-type: none"><li>1. Introdução à disciplina (e-mail e site da disciplina, bibliografia, avaliação).</li><li>2. Motivação para Programar</li><li>3. Hardware, software e princípios</li><li>4. Visão Geral da Linguagem Python</li><li>5. Preparação do Ambiente de Desenvolvimento</li><li>6. Instruções primitivas: atribuição, entrada e saída</li><li>7. Expressões</li><li>8. Tipos</li><li>9. Comandos Condicionais (if)</li><li>10. Tratamento de exceções (try / except)</li><li>11. Exercícios</li><li>12. 1º Teste</li><li>13. Funções</li><li>14. Resolução e entrega do 1º teste</li><li>15. Laços (for e while)</li><li>16. Principais idiomas dos laços</li><li>17. Strings</li><li>18. Arquivos</li><li>19. Listas</li><li>20. Exercícios</li><li>21. 2º Teste</li><li>22. Dicionários</li><li>23. Resolução e entrega do 2º teste</li><li>24. Tuplas</li><li>25. Exercícios</li><li>26. Exercícios</li><li>27. 3º Teste</li></ol>			

- 28. Teste de reposição
- 29. Resolução e entrega do 3º teste
- 30. Resolução e entrega do 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*.
- Web site da disciplina: <http://albertocn.sytes.net/2016-1/icc>
- Ambientes Virtuais de Aprendizagem (AVA) SIGAA e Moodle
- Questionários e Atividades via SIGAA/Moodle
- Questionários com problemas de programação no site <http://thehuxley.com>

#### HORÁRIOS DE ATENDIMENTO

Os horários de atendimento do professor e dos monitores serão divulgados 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

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 ICC tem conteúdo inerentemente acumulativo, o teste de reposição englobará todo o assunto da disciplina.

#### REFERÊNCIAS BIBLIOGRÁFICAS

##### 1. REFERÊNCIAS BÁSICAS:

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