



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

PLANO DE CURSO

IDENTIFICAÇÃO			
Disciplina	Introdução à Ciência da Computação	Código/Turma	COMP0100-T01 COMP0208-T02
Pré-requisito(s)		Carga horária	60
PEL	4.0.0	Créditos	4
Professor(es)	Alberto Costa Neto Carlos Alberto Estombelo Montesco Kleber Tarcísio Oliveira Santos	Semestre	2016.1
EMENTA			
Conceitos gerais. Algoritmos e fluxogramas. Programação científica. Funções e procedimentos.			
OBJETIVOS			
<p>1. GERAL: Apresentar os conceitos básicos e principais técnicas de desenvolvimento de programas de computador, tornando-o apto a compreendê-los e aplicá-los.</p> <p>2. ESPECÍFICOS:</p> <ul style="list-style-type: none">• 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 PROGRAMÁTICO			
<ol style="list-style-type: none">1. Introdução à disciplina (e-mail e site da disciplina, bibliografia, avaliação).2. Motivação para Programar3. Hardware, software e princípios4. Visão Geral da Linguagem Python5. Preparação do Ambiente de Desenvolvimento6. Instruções primitivas: atribuição, entrada e saída7. Expressões8. Tipos9. Comandos Condicionais (if)10. Tratamento de exceções (try / except)11. Exercícios12. 1º Teste13. Funções14. Resolução e entrega do 1º teste15. Laços (for e while)16. Principais idiomas dos laços17. Strings18. Arquivos19. Listas20. Exercícios21. 2º Teste22. Dicionários23. Resolução e entrega do 2º teste24. Tuplas25. Exercícios26. Exercícios27. 3º Teste			

- 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