

Sobre a Disciplina Programação Imperativa

Prof. Alberto Costa Neto

Prof. Breno Piva Ribeiro

Prof. Elisrenan Barbosa da Silva

Prof. Giovanny Fernando Lucero Palma

Prof. Isac Alencar Rodrigues da Silva

Prof. Kalil Araujo Bispo

Prof. Thiago Xavier Rocha de Souza

DComp/UFS



Sobre a Disciplina PI

- **Disciplina:** Programação Imperativa (COMP0334)
- **Equivalentes:**
 - Introdução à Ciência da Computação
 - Programação Imperativa (COMP0197)
- **Carga horária:** 60 horas
- **Créditos:** 4

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

Geral

- Apresentar os conceitos básicos e principais técnicas de desenvolvimento de programas de computador, tornando-o apto a compreendê-los e aplicá-los.

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 Programático

1º Unidade

- Motivação para Programar
- Hardware, software e princípios
- Visão Geral da Linguagem Python
- Preparação do Ambiente de Desenvolvimento
- Instruções primitivas: atribuição, entrada e saída
- Expressões
- Tipos
- Comandos Condicionais (if)
- Tratamento de exceções (try / except)
- Laço While

- Strings
- Laço For
- Funções
- Recursividade
- Listas
- Dicionários
- Tuplas
- Arquivos

2º Unidade

- Desenvolvimento de Projeto

Afinal, por que o nome PI?

- Vem da denominação do Paradigma que vamos estudar: Paradigma Imperativo
 - Você escreve explicitamente as ordens e o computador obedece
 - Mais próximo do funcionamento real do computador
 - Existem outros paradigmas, como por exemplo:
 - Funcional
 - Orientado a Objetos

Método de Ensino

Inovação na Disciplina de PI

- Queremos oferecer um curso melhor
- Usar ferramentas modernas de apoio pedagógico
- Aproveitar a característica da nova geração estar sempre conectada à Internet
- E sobretudo com um *Smartphone* sempre à mão

Metodologia – Ensino Remoto

- Conteúdo teórico estará disponível pela Internet
- Sistema que permite programar e tem autoavaliação
- Tempo de aula (online) dedicado a resolver exercícios e dúvidas

Recursos didáticos e AVA's

Recursos Didáticos

As aulas serão ministradas por meio da Internet, utilizando um software de videoconferência, com horários agendados previamente. As ferramentas utilizadas serão:

- **Ferramentas de Videoconferência:** Google Meet
- **Editores de programas:** Repl.it, 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 Google Classroom
- **Questionários e Atividades** via SIGAA
- **Questionários com Problemas de Programação** no site <http://thehuxley.com>

Ambiente Virtual de Aprendizagem (AVA)

- AVA é um ambiente em rede utilizado para dar apoio ao processo de ensino e aprendizagem tanto na educação presencial como na a distância.
- Nas turmas utilizaremos o próprio SIGAA e o Classroom.
- Nestes ambientes o aluno terá **acesso a todo o conteúdo e realizará atividades** (exercícios, questionários e outros).
- Também podem participar de **fóruns** e se **comunicar** com outros alunos e professores.
- Os professores podem **acompanhar** o desempenho dos alunos.

Correção de Questões

- Imagine se seu professor terá como corrigir 100 questões de cada um dos 50 alunos... Façamos as contas:
 - São 5.000 questões!
 - Supondo que o professor gaste 6 min por questão, seriam necessários 30.000 minutos, ou seja, 500 horas!
- Seria interessante ter uma ferramenta que ajudasse o professor, concordam?



Fonte:
http://2.bp.blogspot.com/_Q4jxiezF5Hk/TNbebADQ2FI/AAAAAAAAABM/gnjeS8-S2I0/s1600/estres-laboral-y-enfermedad-periodontal.jpg

- Uma ferramenta Web que oferece um **banco de problemas de programação** (juiz *on-line*).
- Os **alunos podem enviar soluções** (programas em várias linguagens de programação).
- O **The Huxley executa a solução** com entradas presentes em casos de teste e compara com o resultado esperado.
- Com esta ferramenta o aluno tem um **feedback imediato**

The Huxley



Avaliação

Critério de Avaliação

Através de atividades, teste e projeto, obedecendo à fórmula:

$$\text{Nota Final} = (\text{NOTA1} + \text{NOTA2}) / 2$$

Onde:

- **NOTA1** = **PT** (peso 1) + **EP** (peso 3) + **PP1** (peso 2) + **PP2** (peso 4)
 - **PT**: Prova teórica com questões de múltipla escolha.
 - **EP**: Exercícios de Programação no The Huxley
 - **PP1**: Prova Prática 1
 - **PP2**: Prova Prática 2
- **NOTA2** = Nota do Projeto

Calendário de Avaliações

- As avaliações (atividades, teste e projeto) serão realizadas conforme **calendário** e **orientações** divulgados nos AVA's

Controle de Frequência

Controle de Frequência (Ensino Remoto)

Conforme previsto na resolução Nº 26/2020/CONEPE:

Art. 7º ...

§3º Devido às características inerentes ao ensino remoto, a frequência não será considerada como critério de aprovação, no componente curricular.

Portanto: Não haverá controle de frequência

Bibliografia

Referências Bibliográficas (Básica)

- **Fundamentos da Programação de Computadores.** Ana Fernanda Gomes Ascencio / Edilene Aparecida Veneruchi De Campos. 3º 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 for Everybody: Exploring Data Using Python 3.** Charles R. Severance. CreateSpace Independent Publishing Platform; 1st. ed., 2016; ISBN: 978-1530051120

Referências Bibliográficas (Complementares)

- **Python for Everybody: Exploring Data Using Python** Charles R. Severance. CreateSpace Independent Publishing Platform; 1st. ed., 2016
- **Python for Informatics: Exploring Information**. Charles R. Severance. CreateSpace Independent Publishing Platform; 1st. ed., 2013
- **Como pensar como um Cientista da Computação usando Python (traduzido)**. Allen Downey, Jeffrey Elkner, and Chris Meyers. 2002.
- **Introdução à Programação com Python**. Nilo Ney Coutinho, 2^o edição, 2014, ISBN: 978-85-7522-408-3.
- **Python para Desenvolvedores**. Luiz Eduardo Borges. Rio de Janeiro; 2010
- **Learning to Program Using Python**. Cody Jackson. CreateSpace Independent Publishing Platform

Contatos dos Professores

- Alberto Costa Neto (albertocn@academico.ufs.br) - T11
- Breno Piva Ribeiro (brenopiva@academico.ufs.br) - T03
- Elisrenan Barbosa da Silva (elisrenan@gmail.com) - T01
- Giovanny Fernando Lucero Palma (giovanny@dcomp.ufs.br) - T15 e T20
- Isac Alencar Rodrigues da Silva (isac.silva@dcomp.ufs.br) - T09
- Kalil Araujo Bispo (kalil@dcomp.ufs.br) - T19
- Thiago Xavier Rocha de Souza (txrsouza@gmail.com) - T04 e T10

Como proceder em caso de dificuldade?

- Sempre que identificar alguma dificuldade, dúvida sobre conceitos das videoaulas ou problemas, **entre em contato com o(s) professor(es)** responsáveis pela sua turma.
- Caso não consiga **acessar os AVAs ou sites**, também entre em contato com o(s) professor(es).

Não deixe de tirar suas dúvidas!

E sejam bem-vindos ao curso de PI!!!