

Sobre a Disciplina Programação Imperativa

Prof. Alberto Costa Neto

Prof. Giovanni Fernando Lucero Palma

Prof. Kalil Araujo Bispo

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)
- Funções

2º Unidade

- Laço While
- Strings
- Laços For
- Listas

3º Unidade

- Busca e Ordenação
- Matrizes
- Recursividade
- Dicionários
- Tuplas
- Arquivos

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 - Presencial

- Conteúdo teórico estará disponível pela Internet
- Sistema que permite programar e tem autoavaliação
- Tempo de aula será focado em exercícios e tirar dúvidas

Aulas Presenciais e Horários de atendimento

- As aulas presenciais e os horários de atendimento servirão para **tirar dúvidas e resolver exercícios**
- Sempre que possível, serão alocadas em **laboratório**
- Os professores irão comparecer às aulas (nas turmas presenciais) para:
 - Tirar **dúvidas referentes ao assunto visto nas videoaulas**
 - Tirar **dúvidas e resolver problemas do The Huxley em sala**
 - Levar soluções incompletas (pelo próprio The Huxley, via Pen Drive ou outro meio), para, com o auxílio do professor, fazer correções e submeter ao The Huxley

Recursos didáticos e AVA's

Recursos Didáticos

As aulas serão ministradas presencialmente utilizando as seguintes ferramentas:

- **Youtube**, para exposição das videoaulas.
- **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 com **Problemas de Programação** no site <http://thehuxley.com>

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

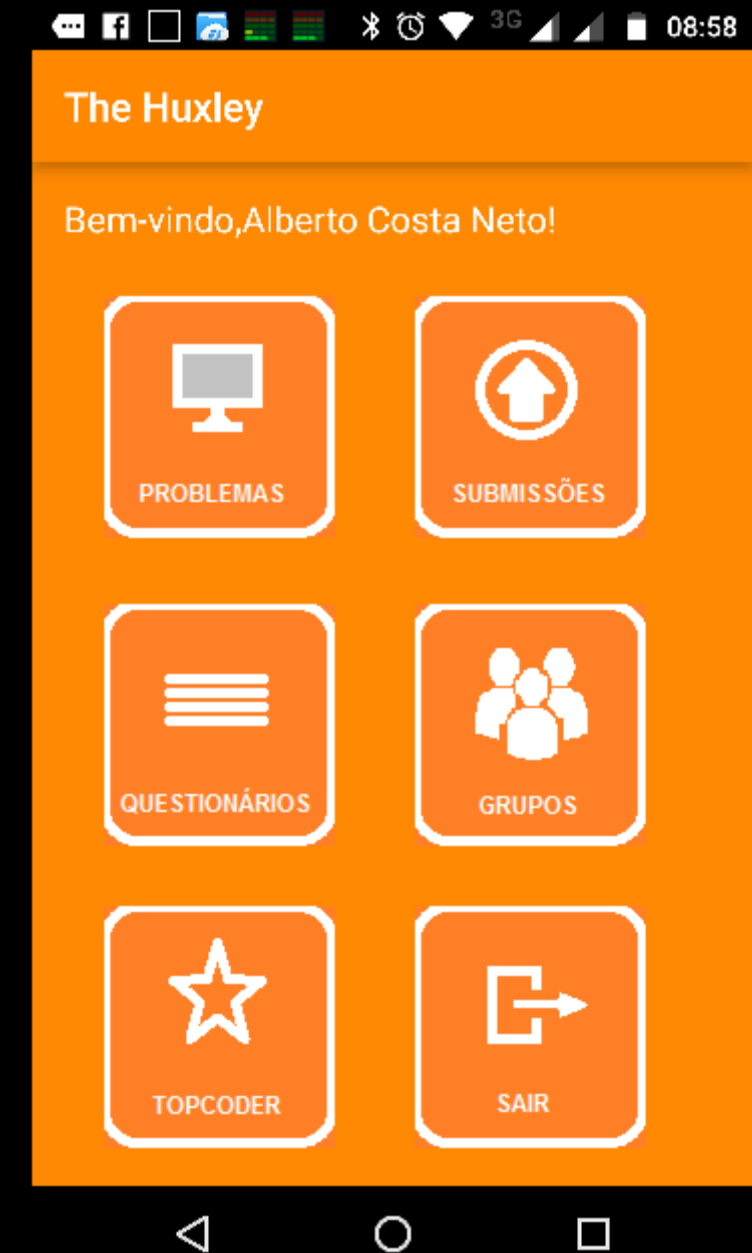
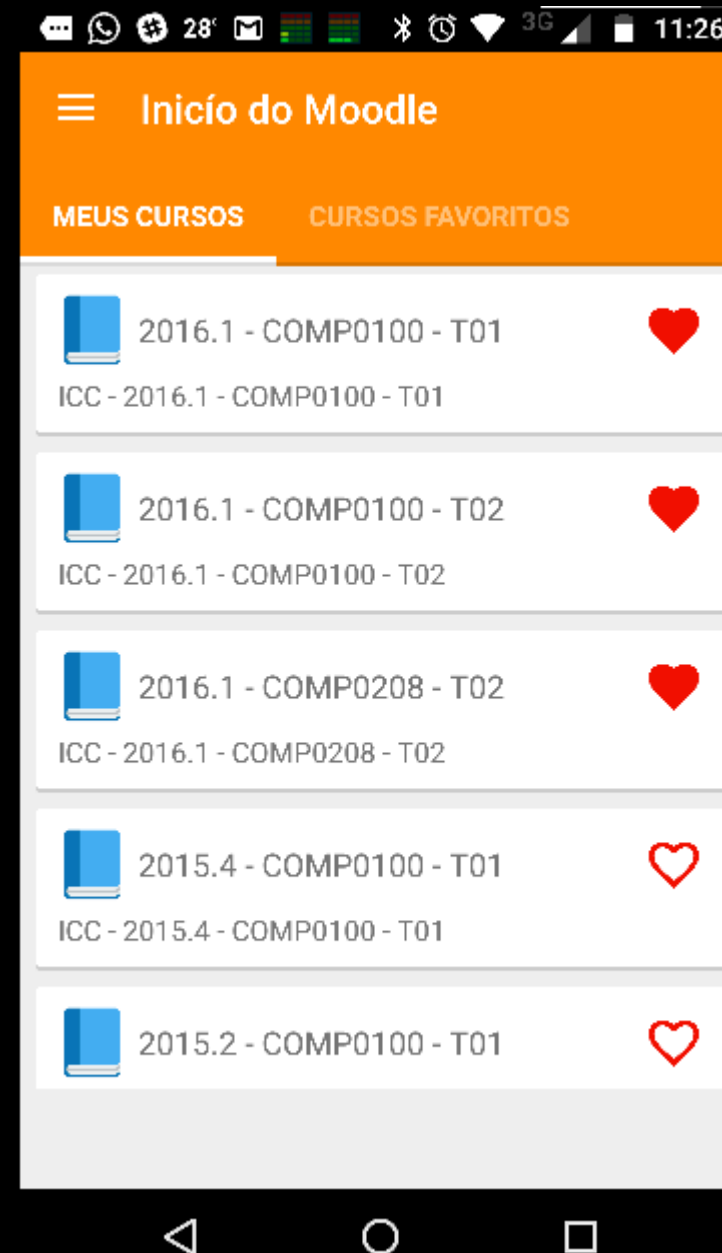
- Uma ferramenta Web que oferece um **banco de problemas de programação** (juiz *on-line*).
- Os **alunos podem enviar soluções** (programas em várias linguagens de programação).
- O **The Huxley executa a solução** com entradas presentes em casos de teste e compara com o resultado esperado.
- Com esta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

The Huxley



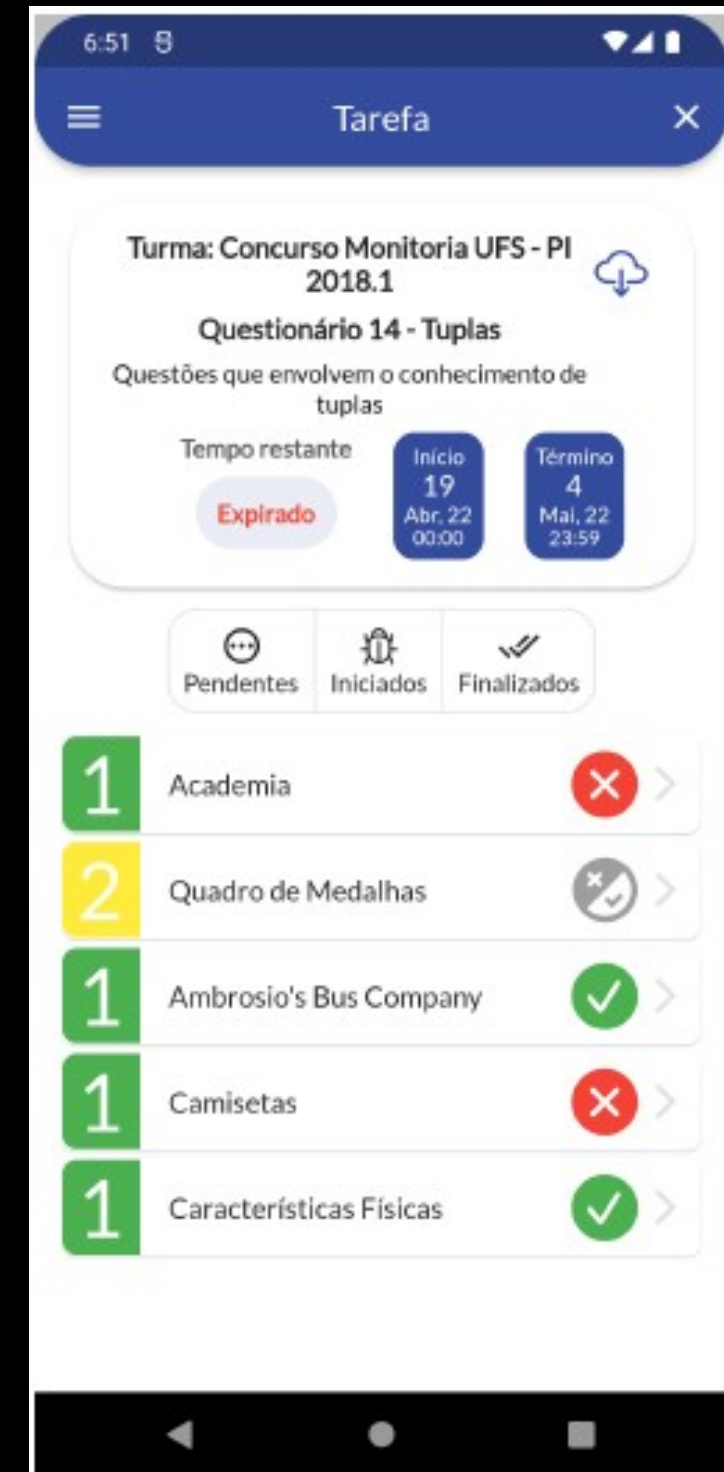
Moodle

- Uma AVA móvel para plataforma Android
- Integra Moodle e The Huxley
- Desenvolvido e mantido por alunos da UFS
- Disponível na Play Store



The Huxley Mobile

- Uma interface para o The Huxley para dispositivos móveis
- Desenvolvido e mantido por alunos da UFS
- Disponível na Play Store



Avaliação

Critério de Avaliação

Através de testes presenciais, obedecendo à fórmula:

$$\text{Nota Final} = (N1 + N2 + N3) / 3$$

Onde:

- **NT1** = Nota do 1º Teste (peso 9) + média dos microtestes (peso 1)
- **NT2** = Nota do 2º Teste (peso 9) + média dos microtestes (peso 1)
- **NT3** = Nota do 3º Teste (peso 9) + média dos microtestes (peso 1)

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.

Calendário de Provas

As provas serão realizadas **presencialmente**:

- No horário da aula
- Segundo **calendário** e **orientações** divulgados nos AVA's

Controle de Frequência

Controle de Frequência (Turmas Presenciais)

- O aluno é obrigado a estar presencialmente nas aulas.
- Assim, a frequência dos alunos será computada através da **Lista de presença**.

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@dcomp.ufs.br) – T06
- Giovanny Fernando Lucero Palma (giovanny@dcomp.ufs.br) – T15 e T16
- Kalil Araujo Bispo (kalil@dcomp.ufs.br) – T04

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!!!