



Disciplina: 1411180 - LABORATÓRIO DE PROGRAMAÇÃO I

Turma: 03 - Período: 2015.2

Ofertada por: 14110000 - UNID. ACAD. DE SISTEMAS E COMPUTAÇÃO

Créditos: 4 - CH: 60

Professores:

- JOÃO ARTHUR BRUNET MONTEIRO

## PLANO DE CURSO

### EMENTA

Conceitos básicos de linguagens de programação. Algoritmos. Noções básicas de programação. Resolução de problemas. Tipos de dados. Entrada e Saída. Decisões. Laços. Funções. Noções de estruturas de dados: listas, conjuntos, vetores, matrizes e mapas. Técnicas básicas de boa programação. Tratamento de erros. Testes.

### I - OBJETIVOS

O objetivo do curso é capacitar o estudante a programar computadores em nível considerado introdutório de acordo com o projeto pedagógico do CCC/UFCG. No curso, adotamos a linguagem de programação Python. A disciplina foca nos conceitos básicos e mais gerais de programação, tais como fundamentos de algoritmos, técnicas e estruturas de dados fundamentais e resolução de pequenos problemas de programação. Ao final do curso, o estudante deve ser capaz de resolver problemas cujas soluções requeiram a escrita de pequenos programas, tipicamente, com dimensões não excedendo uma centena de linhas de código, e que utilizem algoritmos e estruturas de dados fundamentais. Também se espera que o estudante domine as várias tecnologias relacionadas à programação de computadores, tais como operação básica do sistema operacional e ferramentas de suporte à programação. Na disciplina não focamos aspectos avançados e/ou específicos da linguagem, nem mesmo nos conceitos de orientação a objetos (tema que é abordado no segundo semestre, no CCC).

### II - CONTEÚDO PROGRAMÁTICO

O conteúdo do curso foi organizado em 10 unidades, agrupadas em duas partes. A parte 1, que engloba as seis primeiras unidades, reúne os conceitos fundamentais de programação e de linguagens de programação. Nele, esperamos que o estudante desenvolva as habilidades para escrever pequenos programas que lidam com padrões simples de entrada e saída, tipos básicos da linguagem (numéricos, strings, booleans e listas simples), alternativas, condicionais, laços e funções. A parte 2 do curso, formada pelas unidades de 7 a 10 e pelo projeto, aprofunda o uso dos conceitos de linguagem vistos na parte 1, através do estudo de algoritmos e estruturas de dados um pouco mais avançados (tuplas, listas, conjuntos, matrizes, mapas e conjuntos). Na parte 2 também esperamos que o estudante desenvolva mais conscientemente suas habilidades e técnicas de compreensão e resolução de problemas. Finalmente, a parte 2 inclui um pequeno projeto de desenvolvimento, em que o estudante terá a oportunidade de desenvolver um programa de forma mais livre e em equipe.

Descrição das unidades:

Unidade 1 - Conceitos elementares de programação. Conceitos de linguagem de programação abordados: conceito de tipo de dado, tipos numéricos (floats e inteiros), tipo string, variáveis, expressões e avaliação de expressões, invocação de funções matemáticas, formatação básica de strings, comandos de entrada e de saída.

Unidade 2 - Escrevendo programas simples. Conceitos de linguagem abordados: os programas/problemas devem explorar e reforçar os mesmos conceitos de linguagem vistos na unidade 1.

Unidade 3 - Condições, alternativas e funções. Conceitos de linguagem abordados: tipo boolean, expressões lógicas, exit(), comandos condicionais (if, else, elif), introdução a funções, uso e definições de funções simples (sem efeito colateral).

Unidade 4 - Laços definidos. Conceitos de linguagem abordados: strings como sequências, listas, indexação de sequências, operador in, laços for, função range, iteradores (laços sobre valores) versus laços sobre índices.

Unidade 5 - Laços indefinidos. Conceitos de linguagem de programação: comando while, break e continue. Busca linear com while e com for.

Unidade 6 - Funções. O foco será a compreensão da semântica de funções, bem como abstração funcional como técnica de design. A sintaxe e a semântica devem ser expressas como uma combinação de texto e de casos de teste (asserts). Conceitos de linguagem: invocação de funções, definição de funções, parâmetros formais, parâmetros reais, retorno, pilha de execução, semântica de execução de funções, escopo, funções com efeito colateral vs. funções puras, aliasing, stack frames.

Unidade 7 - Algoritmos com laços. Os problemas abordados nesta unidade envolvem formas básicas de manipulação de listas. Em particular, serão abordados os padrões inserção ordenada, merge e algoritmos básicos de ordenação (bubble sort, insertion sort, selection sort).

Unidade 8 - Estruturas de dados: sequências. Conceitos de linguagem de programação: slices, métodos e construtores de listas e tuplas (+, append, insert, pop, remove, etc).

Unidade 9 - Estruturas de dados: matrizes.

Unidade 10 - Estruturas de dados: mapas. Conceito de linguagem: sintaxe e semântica de mapas.

### III - METODOLOGIA

O curso é baseado em um método de ensino e aprendizagem conhecido como mastery learning. Nesse método, o conteúdo (conhecimento + habilidades) do curso é organizado em unidades que o estudante deve abordar sequencialmente, progredindo entre unidades somente quando demonstrar ter dominado o conteúdo e as habilidades da unidade anterior. Para avaliar se o estudante domina o conhecimento e as habilidades requeridas em cada unidade, usamos minitestest semanais. É importante enfatizar: o estudante não poderá avançar para a unidade N, enquanto não for aprovado na unidade N-1. E para ser aprovado em uma unidade, deve ser capaz de resolver pelo menos dois problemas diferentes da unidade.

### IV - AVALIAÇÃO

A avaliação será baseada no progresso do aluno nas unidades, o desempenho obtido nas avaliações automáticas (funcional) e qualitativa dos minitestest. Os pesos de cada componentes são os seguintes:

Minitestest - 85%

Avaliação Qualitativa - 15%

### V - REFERÊNCIAS BIBLIOGRÁFICAS

- Think Python, de Allen Downey (<http://www.greenteapress.com/thinkpython/>). Há uma versão em português do livro. Observe, contudo, que é uma tradução de uma versão já antiga.

- Python Programming: An Introduction to Computer Science, de John Zelle (<http://mcsp.wartburg.edu/zelle/python/>). Alguns capítulos deste livro são excelentes na explicação dos conceitos de programação.

- Dive into Python (<http://www.diveintopython.net/>). Este é um livro para quem já programa e quer aprender Python. Talvez não seja o caso para vários dos nossos alunos. Está aqui apenas, para registro.