



Ficha 2 (variável)

Disciplina: Fundamentos de Programação de Computadores (2023-1)					Código: CI182 / CI240		
Natureza: (X) Obrigatória () Optativa		(X) Semestral () Anual () Modular					
Pré-requisito:		Co-requisito:	Modalidade: (X) Presencial () Totalmente EaD () ____ *C.H.EaD				
CH Total: 60h CH semanal: 4h		Padrão (PD): 60	Laboratório (LB): 00	Campo (CP): 00	Estágio (ES): 00	Orientada (OR): 00	Prática Específica (PE): 00
		Estágio de Formação Pedagógica(EPP): 00	Extensão (EXT): 00	Prática como Componente Curricular (PCC): 00			
EMENTA (Unidade Didática)							
Algoritmos e Estrutura de dados básicos. Introdução a uma linguagem de programação. Solução de problemas simples de manipulação de dados utilizando o computador.							
PROGRAMA (itens de cada unidade didática)							
CONTEÚDO							
1	Apresentação da disciplina: Apresentação das regras, datas de avaliações, critério de notas, e apresentação da bibliografia.						
2	Introdução ao conceito de algoritmos e à linguagem de programação Python. Uso de ambiente de desenvolvimento de programas em Python.						
3	Estrutura básica de um programa em Python. Sentenças. Variáveis e tipos de dados. Comandos simples, comandos de atribuição. Expressões aritméticas. Comandos de entrada (leitura) e saída (escrita). Funções matemáticas. Exemplos de Programas.						
4	Expressões relacionais e lógicas. Estruturas de desvio condicional.						
5	Estruturas de Repetição.						
	Prova 1						
6	Funções: conceito e estrutura de definição e uso.						
7	Conceito de vetores e matrizes. Estruturas de dados fundamentais: listas e tuplas.						
	Prova 2						
8	Manipulação de entrada e saída de dados. Uso de arquivos.						
9	Uso de bibliotecas de funções						
	Prova 3						



OBJETIVO GERAL

Apresentar ao aluno técnicas elementares de programação e de desenvolvimento de algoritmos. Ao final da disciplina o aluno deve ser capaz de implementar programas para ler dados em formato texto, efetuar operações simples sobre estes dados e imprimir ou gravar o resultado em arquivos.

OBJETIVO ESPECÍFICO

1. O aluno será capaz de entender o conceito básico de algoritmo e a sua relação com o que se denomina Programa de Computador.
2. O aluno será capaz de criar Programas que obtém valores numéricos [(reais ou inteiros) do usuário, realizam algum cálculo aritmético com estes valores e exibem o resultado na tela do computador.
3. O aluno será capaz de criar programas em que existem alternativas de cálculos ou solução de um certo problema conforme condições estabelecidas em valores numéricos lidos ou calculados pelo programa.
4. O aluno será capaz de criar programas em que determinados cálculos ou operações de leitura e escrita se repetem enquanto uma certa condição ou conjunto de condições forem verdadeiras. (por exemplo, obter do usuário um conjunto de 20 valores numéricos e com cada um deles executar um cálculo específico, imprimindo o resultado em cada vez.).
5. O aluno será capaz de criar programas em que seja necessário armazenar ou obter um conjunto de valores numéricos para então proceder a cálculos com o conjunto de valores lidos. Problemas que envolvem a análise básica de cadeias caracteres ou situações similares serão possíveis de serem resolvidos.
6. O aluno será capaz de criar programas capazes de ler dados de um arquivo em disco, efetuar operações e cálculos com visto nos itens anteriores e depois gravar os resultados em arquivos novamente.
7. O aluno será capaz de utilizar algumas funções pré-definidas pela linguagem Python usadas para a apresentação de dados ou efetuar cálculos mais complexos.

PROCEDIMENTOS DIDÁTICOS

1. **Atividades síncronas:** As atividades síncronas consistirão de aulas **presenciais** em sala, com duração total de **2 horas** por aula.
2. **Material didático específico:** Serão utilizados vídeos e documentos digitalizados como material de referência básico sobre o tema da disciplina.
3. **Infraestrutura de suporte tecnológico, científico e instrumental à disciplina:** Os exercícios práticos de programação poderão ser executados em computadores e *smartphones*, com a utilização de ambientes de programação com licença de uso livre e disponíveis para os principais sistemas operacionais.
4. **Previsão de período de ambientação dos recursos tecnológicos a serem utilizados pelos discentes:** Haverá na primeira semana de aula a disponibilização de material de leitura indicando como deverá ser o andamento da disciplina. Neste período, o professor colocará em site específico as orientações para acesso à página principal da disciplina, onde ficarão disponibilizados datas de provas, materiais de estudo, exercícios de programação, e outras informações de apoio.
5. **Identificação do controle de frequência das atividades:** O controle de frequência será feito com base no comparecimento do aluno nas aulas presenciais.

FORMAS DE AVALIAÇÃO

Deverão ser feitas 3 (três) provas (atividade **presencial** em sala de aula).

Em quaisquer atividades avaliativas (provas e exercícios), se forem constatadas similaridades e plágio, os alunos envolvidos serão chamados pelo professor e poderão receber nota 0 (zero), conforme regimentos vigentes na UFPR.

Provas não realizadas pelo aluno são passíveis de 2ª-chamada, nos casos amparados pelo artigo 106, Seção V, Resolução 37/97-CEPE, e considerando também o disposto no artigo 12, § 7º e 8º, Resolução 22/21-CEPE, em data e local divulgados no Cronograma da disciplina.

As médias parcial (**MP**) e final (**MF**) serão calculadas da seguinte forma, de acordo com os critérios para aprovação com ou sem exame final seguirão o disposto na Resolução 37/97-CEPE. Capítulo X, Seção I – Normas Gerais de Avaliação:

$$MP = 0,3 \times P1 + 0,3 \times P2 + 0,4 \times P3$$

Se $MP \geq 70$ → Aprovado, com $MF = MP$

Se $MP < 40$ → Reprovado por nota

Se $MP \geq 40$ → Exame Final :

$$MF = (MP + EXAME) / 2$$

Se $MF < 50$ → Reprovado por nota

BIBLIOGRAFIA BÁSICA (mínimo 03 títulos)

- [1] Lutz, M., Ascher D. Aprendendo Python. Bookman, 2007.
- [2] Marco Medina e Cristina Fertig. Algoritmos e Programação: Teoria e Prática. 2a. edição. Novatec Editora Ltda., 2006.
- [3] Menezes, N.N.C. Introdução à Programação com Python, Novatec, 2010.
- [4] Downey, A. B., Pense em Python (tradução da segunda edição do livro Think Python). Disponível em <https://penseallen.github.io/PensePython2e/>, 2016.
- [5] Wentworth, P., Elkner, J., Downey, A. B., Meyers, C. How to think like a computer scientist: Learning with Python 3. Disponível em: <http://openbookproject.net/thinkcs/python/english3e/>, 2012

BIBLIOGRAFIA COMPLEMENTAR (mínimo 05 títulos)

- [1] Raul, W. Introdução a Algoritmos e Programação com Python - Uma Abordagem Dirigida Por Testes. Grupo GEN, 2017.
- [2] Banin, S. L. Python 3 - Conceitos e Aplicações - Uma abordagem didática. Editora Saraiva, 2018.
- [3] Ljubomir, P. Introdução à Computação Usando Python - Um Foco no Desenvolvimento de Aplicações. Grupo GEN, 2016.
- [4] HETLAND, M.L., Beginning Python: From Novice to Professional, Springer-Verlag, 2005, ISBN:1-59059-519-X
- [5] LEE, K.D., HUBBARD, S., Data Structures and Algorithms with Python. 2015.
- [6] Donald E. Knuth. The Art of Computer Programming. 1997.
- [7] Sweigart, A. Making Games with Python & Pygame, <http://inventwithpython.com/pygame/>, 2012.



Ministério da Educação
UNIVERSIDADE FEDERAL DO PARANÁ
Setor de Ciências Exatas
Departamento de Informática

Professores da Disciplina: Prof. Armando Luz N Delgado

Assinatura: _____

Chefe de Departamento ou Unidade equivalente: Prof. Dr. Luis Carlos Erpen de Bona

Assinatura: _____