

Ficha 2 (variável)

Disciplina: Fundamentos de Programação de Computadores (2021-2)					Código: CI182/CI240	
Natureza: (X) Obrigatória () Optativa		(X) Semestral () Anual () Modular				
Pré-requisito:		Co-requisito:		Modalidade: (X) Presencial () Totalmente EaD () ____ *C.H.EaD () 100% ERE (Ensino Remoto Emergencial), Res. 22/21-CEPE		
CH Total: 60h CH semanal: 4h		Padrão (PD): 60	Laboratório (LB): 00	Campo (CP): 00	Estágio (ES): 00	Orientada (OR): 00
		Estágio de Formação Pedagógica(EPP): 00	Extensão (EXT): 00	Prática como Componente Curricular (PCC): 00	Prática Específica (PE): 00	

EMENTA (Unidade Didática)

Histórico e paradigmas de linguagens de programação. A linguagem de programação Python (v3). Variáveis e tipos de dados. Componentes de um programa. Programação procedural estruturada em blocos. Estruturas condicionais e de repetição. Vetores, listas, tuplas. Entrada e Saída. Exemplos de algoritmos.

PROGRAMA (itens de cada unidade didática)

Aulas	Datas	CONTEÚDO
1-2	31/jan-04/fev	Apresentação do disciplina: Definição das regras, provas, notas, apresentação da bibliografia. Ambientação dos alunos ao Moodle e recursos a serem usados durante a disciplina. Introdução a algoritmos.
3-4	07/fev-11/fev	Linguagens de programação: histórico e paradigmas. A linguagem de programação Python 3. Ferramentas para apoio à programação em Python.
5-6	14/fev-18/fev	Variáveis e tipos de dados. Blocos de comandos.
7-11	21/fev-11/mar	Estruturas condicionais.
12	15/mar	Prova 1 (para turmas com aulas na 3ª- e 5ª-feira)
	16/mar	Prova 1 (para turmas com aulas na 4ª- e 6ª-feira)
13-15	17/mar-25/mar	Estruturas de repetição.
16-19	28/mar-08/abr	Estruturas de dados fundamentais: vetores, listas e tuplas.
20-24	11/abr-27/abr	Funções simples e compostas. Entrada e saída por arquivos e passagem de argumentos.
25	28/abr	Prova 2 (para turmas com aulas na 3ª- e 5ª-feira)
	29/abr	Prova 2 (para turmas com aulas na 4ª- e 6ª-feira)
26	02/mai-06/mai	Finalização da disciplina. Apresentação de trabalhos.
	03/mai	2ª-chamada Provas 1 e 2 (para turmas com aulas na 3ª- e 5ªfeira)
	04/mai	2ª-chamada Provas 1 e 2 (para turmas com aulas na 4ª- e 6ªfeira)
-	12/mai	Exame Final (para turmas com aulas na 3ª- e 5ª-feira)
	13/mai	Exame Final (para turmas com aulas na 4ª- e 6ª-feira)



OBJETIVO GERAL

Capacitar o estudante a compreender algoritmos e aplicar técnicas básicas de programação de computadores na solução de problemas diversos, visando a confecção de seus próprios programas.

OBJETIVO ESPECÍFICO

1. Introduzir o conceito de algoritmo para resolução de problemas e automação de tarefas por meio de exemplos da vida real.
2. Apresentar as linguagens de programação e seus paradigmas, expondo brevemente as peculiaridades e aplicações de cada um deles.
3. Apresentar a linguagem de programação Python 3, seu objetivo, histórico de desenvolvimento, características, vantagens e desvantagens em relação a outras linguagens existentes.
4. Mostrar diferentes ferramentas para programação em Python com recursos gráficos e em linha de comando, e ensinar a configuração de um ambiente para programação e execução de códigos na linguagem.
5. Explicar o que é um programa, quais são seus elementos estruturantes, o que são e para que servem as variáveis e os tipos de dados, e como utilizá-los em um programa.
6. Mostrar os comandos aceitos por uma linguagem de programação, bem como criar blocos deles para executar ações e implementar algoritmos.
7. Apresentar as estruturas condicionais e sua aplicação na mudança do fluxo de um programa. Comandos IF, ELIF, ELSE e aninhamento.
8. Apresentar as estruturas de repetição e sua aplicação em programas que necessitam de laços de iteração. Comandos FOR, WHILE e aninhamento.
9. Ensinar o conceito de função no âmbito de programas computacionais, passagem de parâmetros, retorno de valores, chamadas e composição.
10. Introduzir estruturas de dados fundamentais de armazenamento de informações e suas aplicações: vetores para dados de um tipo, listas para dados variados, tuplas para imutabilidade.
11. Apresentar a entrada e saída de dados por meio da chamada do programa (linha de comando), arquivos externos ou funções com argumentos.

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 documentos digitalizados como material de referência básico sobre o tema da disciplina. Também serão disponibilizados links para sites existentes para exercitar os conceitos básicos e eventualmente materiais já disponíveis em MOOC's (Cursos Massivos Abertos). O professor também poderá produzir vídeos próprios onde serão esclarecidos aspectos específicos ou avançados que possam surgir no decorrer 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:** 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 utilizando como apoio a UFPR Virtual. Neste período, o professor enviará por e-mail as orientações para acesso inicial ao Moodle e à



página da disciplina neste ambiente. Haverá também neste momento uma atividade síncrona de forma a esclarecer dúvidas e ajudar na solução dos eventuais problemas que possam ocorrer com o acesso on-line e instalação de programas.

5. **Identificação do controle de frequência das atividades:** O controle de frequência será feito com base no comparecimento do aluno nas aulas presenciais e na entrega de **Exercícios de Programação (EP), dentro do prazo**. O cumprimento das metas estabelecidas nestes exercícios contarão como presença, **desde que haja completude e coerência com os conteúdos da disciplina e com o que é solicitado nesta atividade**.

- As aulas presenciais corresponderão a **26 aulas de 2 horas = 52 horas** (de acordo com resoluções **22/21-CEPE** e **52/21-CEPE**, e instrução normativa **IN 02/21-PROGRAD**)
- Os exercícios de programação (**EP**) corresponderão, em seu conjunto, a **8 horas**, sendo a frequência assim contabilizada:

$$8 * (\text{Quantidade EP's entregues} / \text{Quantidade total EP's})$$

6. **Carga horária semanal:** As atividades serão distribuídas da seguinte forma:

- Atividades presenciais: **26 aulas** com **2h/aula** o que totaliza **52h** presenciais no semestre.
- Exercícios de Programação (**EP**): **8h/semestre** de atividade fora do horário de aula presencial. O professor definirá um horário de atendimento presencial na UFPR para atendimento dos alunos durante o desenvolvimento destes exercícios.

FORMAS DE AVALIAÇÃO

Deverão ser feitas 2 (duas) provas (atividade **presencial** em sala de aula) e haverá um conjunto de exercícios de programação (**EP**) que deverão ser entregues pelo aluno dentro de um prazo estipulado.

Para a validação dos exercícios, os alunos poderão ser chamados, a critério do professor, em um momento presencial nos horários de atendimento do professor. Serão usados sistemas de detecção de similaridade nas produções dos alunos.

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.

Os exercícios citados anteriormente deverão ser entregues dentro do prazo estipulado nos respectivos enunciados. A nota final de exercícios (**EXERCS**) será a média aritmética das notas de todos os exercícios de programação (**EP**). Exercícios não entregues terão nota 0 (zero).

O comparecimento do aluno em aula presencial e a entrega dos exercícios de programação (**EP**) serão contabilizados na frequência conforme indicado no item 5 dos **Procedimentos Didáticos**. A entrega dos exercícios será contabilizada como frequência ao receber a nota do professor.

Não serão aceitas entregas de exercícios após o final do período letivo previsto para a 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:

$$MP = 0,4 \times (P1 + P2) + 0,2 \times EXERCS$$

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

Se $MP < 40$ → Reprovado por nota



Se $MP \leq 40$ → Exame Final :
 $MF = (MP + EXAME) / 2$
Se $MF < 50$ → Reprovado por nota

BIBLIOGRAFIA BÁSICA (mínimo 03 títulos)

- [1] Raul, W. Introdução a Algoritmos e Programação com Python - Uma Abordagem Dirigida Por Testes. Grupo GEN, 2017. 9788595156968. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788595156968/>.
- [2] Banin, S. L. Python 3 - Conceitos e Aplicações - Uma abordagem didática. Editora Saraiva, 2018. 9788536530253. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536530253/>.
- [3] Ljubomir, P. Introdução à Computação Usando Python - Um Foco no Desenvolvimento de Aplicações. Grupo GEN, 2016. 9788521630937. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788521630937/>.
- [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] 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] Donald E. Knuth. The Art of Computer Programming. 1997.
- [5] Sweigart, A. Making Games with Python & Pygame, <http://inventwithpython.com/pygame/>, 2012.

Professores da Disciplina: Prof. Giovanni Venâncio de Souza, Profª Cleide Possamai

Assinatura: _____

Chefe de Departamento ou Unidade equivalente: Prof. Dr. Fabiano Silva

Assinatura: _____



CI182/CI240 - Fundamentos de Programação de Computadores

I. Período atividades: de 31/jan/2022 a 07/maio/2022 [14 semanas]

- 26 aulas presenciais de 2h cada = 52h presenciais
- 8h exercícios de programação (EP)

II. Plano de Ensino e Cronograma da disciplina

Vide **Ficha 2**.

III. Turmas, vagas e professor responsável:

CURSO	TURMA	VAGAS	PROFESSOR	REGIME	Horário aulas presenciais
Estatística	EST1	40	Giovanni	Presencial	4 ^a -feira 19:30h - 21:30h 6 ^a -feira 21:30h - 23:30h

IV. Contato de Professores responsáveis:

Giovanni Venâncio de Souza <gvs11ufpr@gmail.com>

Cleide Possamai <cleidepossamai@gmail.com>