

Curso: CST em Análise e Desenvolvimento de Sistemas		Ano Letivo: 2021/1
Nome da Disciplina: Linguagem de Programação		Turma: D
Carga Horária: 80 horas	Aulas: 100% Práticas	Série: 1º Período
Docente: Profa. Simone de Abreu		
Coordenação: Profa. Ana Cristina dos Santos		
Competências		
Entender o que é Linguagem de Programação, Paradigmas de Programação e diferenciar tipos de linguagem, compilação e interpretação.		
Compreender como a lógica está relacionada aos algoritmos, programas e linguagens de programação.		
Conhecer os principais componentes e estruturas de uma linguagem de programação, como variáveis, tipos primitivos, operadores aritméticos, coleções, estruturas de decisão, estruturas de repetição e módulos.		
Saber compreender um problema e como resolvê-lo pela lógica e linguagem de programação Python.		
Conhecer a sintaxe da linguagem Python, e como escrever os principais componentes e estruturas nesta linguagem.		
Habilidades		
Executar comandos na <i>Shell</i> do IDLE Python para automatizar tarefas.		
Desenvolver algoritmos e programas de computador na linguagem Python, utilizando variáveis, tipos primitivos, operadores aritméticos, coleções, estruturas de decisão, estruturas de repetição, funções e listas lineares.		
Desenvolver a técnica de refinamentos sucessivos na solução de problemas, aprendendo a decompor um problema em sub-tarefas ou funções.		
Disciplinas Relacionadas		
<ul style="list-style-type: none"> • Anteriores: Nenhuma • Paralelas: Lógica de Programação • Posteriores: Programação Orientada a Objetos 		
Conteúdo Programático		
<ul style="list-style-type: none"> • 01 - Introdução, ambiente de desenvolvimento e estrutura básica de Python <ul style="list-style-type: none"> ○ Aula 1 - Introdução à disciplina e conceitos iniciais; ○ Aula 2 - Ambiente de desenvolvimento, tipos de dados e variáveis; ○ Aula 3 - Funções embutidas, entrada e saída, operadores e expressões aritméticas. • 02 - Expressões relacionais e lógicas, estruturas de seleção e funções <ul style="list-style-type: none"> ○ Aula 4 - Operadores e expressões relacionais e lógicas, estruturas de seleção simples e composta; ○ Aula 5 - Estruturas de seleção aninhadas e encadeadas; ○ Aula 6 - Criação de funções e escopo de variáveis. • 03 - Estruturas de repetição <ul style="list-style-type: none"> ○ Aula 7 - Estrutura de repetição <i>while</i>; ○ Aula 8 - Estruturas de repetição aninhadas; 		

- Aula 9 - Sequências e estrutura de repetição *for*.
- 04 - Sequências (*strings*, listas, tuplas e intervalos)
 - Aula 10 - Operações básicas com sequências;
 - Aula 11 - Busca e ordenação em sequências;
 - Aula 12 - Sequências aninhadas.

Metodologia de ensino

- Aulas expositivas utilizando projetor, lousa eletrônica e computador nas quais se apresenta e discute os tópicos da disciplina, bem como trabalhos em grupo com apresentação escrita e defesa oral, apresentação de vídeos;
- Aulas gravadas nas quais os tópicos da disciplina são apresentados e discutidos;
- Indicação de material extra para leitura;
- Atividades contínuas (AC) para acompanhamento do processo de ensino e aprendizagem.

CrITÉRIOS de Avaliação

A avaliação será composta por 3 atividades contínuas (ACs) realizadas ao longo do semestre e uma prova realizada ao final do semestre. Para ser aprovado, o aluno deve obter uma nota final mínima de 6,0.

Critério de aprovação: $Nota\ final \geq 6$

O cálculo da nota final é dado por:

$$Nota\ final = 50\% MAC + 50\% Prova$$

Onde:

- MAC é a média simples de 3 Atividades Contínuas
- Prova é a nota obtida na avaliação realizada ao final do semestre

* O aluno tem direito a uma Prova Substitutiva, com todo o conteúdo do semestre letivo, para substituir a Prova Regular. A nota da Prova Substitutiva somente será utilizada se for maior que a da Prova Regular.

Bibliografia Básica

DIERBACH, C. **Introduction to Computer Science Using Python: A Computational Problem-Solving Focus** 1st Edition, New York: Wiley, 2012.

MENEZES, N. N. C. **Introdução à Programação com Python: algoritmos e lógica de programação para iniciantes**. 2ª. ed. São Paulo: Novatec, 2014.

ZELLE, J. **Python Programming: An Introduction to Computer Science**, 3rd Ed. Sherwood: Franklin, Beedle & Associates Inc.; 2016.

Bibliografia Complementar

CARVALHO, A.P.L.F.; LORENA, A.C. **Introdução à computação: software, hardware e dados**. 1a. ed. Rio de Janeiro: LTC, 2017.

DOWNEY, A. B.; **Pense em Python: Pense como um cientista da computação**. 2. ed. São Paulo: Novatec, 2016.

FARRELL, J. **Lógica e Design de Programação** – Introdução. 1. ed. São Paulo: Cengage Learning, 2010.

LOPES, A.; GARCIA, G. **Introdução à Programação: 500 Algoritmos**. Rio de Janeiro: Elsevier, 2002.

MOKARZEL, F. C.; SOMA, N. Y. **Introdução à Ciência da Computação**. Rio de Janeiro: Elsevier, 2008.

Plano de aulas		
Semana	Data	Conteúdo
1	12-abr	<p>(1) Apresentação do professor.</p> <p>(2) Apresentação da disciplina: (a) organização do Google Classroom; (b) acesso à apostila; (c) acessos às Atividade Contínuas; (d) acesso às gravações; (e) acesso ao material de apoio desenvolvido durante a aula; (f) acesso ao Plano de Ensino; (g) acesso à Prova Regular e Substitutiva no formato remoto; (h) acesso à sala de aula remota no Google Meet; (i) acesso aos vídeos de apoio ao ambiente URI Online Judge.</p> <p>(3) Aprofundamento nos tópicos do Plano de Ensino, como critérios de avaliação, condições para aprovação e bibliografia.</p> <p>(4) Horário das aulas e intervalos.</p> <p>(5) Procedimento para contabilização das presenças.</p> <p>(6) Procedimento para realização das Atividades Contínuas.</p> <p>Apostila, Capítulo 1: (I) introduzir as diretrizes da disciplina, como conteúdo programático, metodologia de ensino, critérios de avaliação e bibliografia; (II) conceituar os termos “algoritmo”, “lógica de programação”, “linguagem de programação”, “código-fonte” e “programa”; (III) simular o processo para resolução de problemas computacionais sem o uso de computadores (computação desplugada).</p>
1	14-abr	<p>Apostila, Capítulo 2: (I) preparar o ambiente computacional necessário para a disciplina; (II) acessar as duas ferramentas básicas utilizadas na disciplina para o desenvolvimento de programas em Python (interpretador interativo e editor de código fonte); (III) conceituar e utilizar tipos de dados primitivos, constantes e variáveis; (IV) conhecer os operadores aritméticos; (V) manipular o sistema <i>online</i> Python Tutor.</p>
2	19-abr	<p>Apostila, Capítulo 3: (I) relembrar expressões; (II) conhecer os operadores de atribuição composta; (III) compreender o conceito de precedência e associatividade de operadores; (IV) entender como Python avalia expressões; (V) descobrir o que são funções e como usar funções integradas; (VI) exibir dados para o usuário; (VII) obter entradas do usuário; (VIII) conhecer tipos de erros comuns em programação.</p> <p>Apostila, Capítulos 1, 2 e 3: Exercícios complementares.</p>
2	21-abr	Feriado (Dia de Tiradentes).
3	26-abr	<p>Apostila, Capítulo 4: (I) apresentar as estruturas de controle de fluxo de execução; (II) conhecer as expressões relacionais e lógicas, assim como seus operadores; (III) entender a associatividade dos operadores relacionais; (IV) compreender o que são expressões equivalentes e complementares; (V) conceituar a avaliação de curto-circuito; (VI) construir estruturas de seleção simples e composta.</p>
3	28-abr	<p>Apostila, Capítulo 5: (I) avançar o estudo de <i>strings</i>; (II) introduzir o conceito de <i>flags</i> booleanas; (III) apresentar as estruturas de seleção aninhadas e encadeadas; (IV) entender o funcionamento do comando <i>elif</i>.</p> <p>AC1 - Entre 28/abril e 02-maio (23:59h)</p>
4	03-mai	Apostila, Capítulos 4 e 5: Exercícios complementares.
4	05-mai	<p>Apostila, Capítulo 6: (I) conhecer as funções importadas; (II) definir nossas próprias funções; (III) entender a relação entre argumentos e parâmetros de funções; (IV) compreender o que é valor de retorno; (V) introduzir o conceito de escopo de variáveis; (VI) escrever documentação nas funções; (VII) organizar o código-fonte.</p>

5	10-mai	Apostila, Capítulo 6: Exercícios complementares.
5	12-mai	Apostila, Capítulo 7: (I) entender a necessidade da estrutura de repetição para controle do fluxo de execução; (II) compreender a estrutura de repetição com quantidade de repetições indefinida: laço while; (III) usar variáveis contadoras, acumuladoras e de sinalização booleana para controlar a execução do while; (IV) conceituar “laço infinito” e suas consequências; (V) utilizar estruturas de repetição combinadas com estruturas de seleção. AC2 - Entre 14/maio e 20-maio (23:59h)
6	17-mai	Apostila, Capítulo 8: (I) aprender os comandos de encerramento e interrupção de laços; (II) simular a estrutura de repetição <i>repeat...until</i> usando while; (III) utilizar laços de repetição para validação de entradas; (IV) entender a necessidade das estruturas de repetição aninhadas; (V) resolver problemas com repetições aninhadas.
6	19-mai	Apostila, Capítulos 7 e 8: Exercícios complementares.
7	24-mai	Apostila, Capítulo 9: (I) compreender o que são sequências; (II) conhecer os tipos de sequências em Python: <i>strings</i> , listas, tuplas e intervalos; (III) distinguir sequências mutáveis e imutáveis; (IV) distinguir sequências homogêneas e heterogêneas; (V) percorrer sequências por meio de seus índices; (VI) entender a estrutura de repetição com quantidade de repetições definida: laço for; (VII) percorrer sequências por meio de seus itens com o laço for.
7	26-mai	Apostila, Capítulo 9: Exercícios complementares. AC3 - Entre 28/maio e 03-junho (23:59h)
8	31-mai	Apostila, Capítulo 10: (I) explorar o funcionamento do Python em relação aos tipos de dados e sua representação em memória; (II) conhecer as principais funções e métodos para trabalhar com sequências; (III) compreender os tipos de passagem de argumentos para funções e como são aplicados em Python.
8	02-jun	Apostila, Capítulo 11: (I) entender como buscar itens em sequências; (II) conhecer e implementar os algoritmos de busca linear e busca binária; (III) construir uma função para ordenação de sequências; (IV) resumir o funcionamento de outros algoritmos de ordenação; (V) conhecer recursos integrados ao Python para ordenação; (VI) aprender como chamar funções com argumentos nomeados e como defini-las com argumentos padrão.
9	07-jun	Apostila, Capítulo 12: (I) conhecer as sequências aninhadas; (II) usar sequências aninhadas para representar matrizes e tabelas; (III) recordar os conceitos de objetos mutáveis e imutáveis; (IV) entender o mecanismo de cópia de objetos em Python; (V) compreender o conceito e uso de cópias rasas e cópias profundas.
9	09-jun	Apostila, Capítulos 10, 11 e 12: Exercícios complementares.
10	14-jun	Prova Regular.
10	16-jun	Vista da Prova Regular.
11	21-jun	Prova Substitutiva.
11	23-jun	Vista da Prova Substitutiva.