DIRETORIA ACADÊMICA

PLANO DE ENSINO

Curso: Análise e Desenvolvimento de Sistemas		
Unidade Curricular: Algoritmos e Lógica de Programação		**Carga Horária:** 80h
Modalidade: Presencial (X) Semipresencial ( ) EAD ( )		
Período: 1º	**Ano/ Semestre Letivo:** 2026.1	
Professor (es): Walter Travassos Sarinho		

1. EMENTA

Conceitos básicos de algoritmos. Comandos de entrada e saída de dados. Tipos de dados e operações primitivas. Variáveis e constantes. Estruturas de seleção. Estruturas de repetição. Estrutura de dados compostas, modularização, implementadas em linguagem de alto nível, usando noções de estruturação de código. Documentação.

2. OBJETIVO (S)

2.1. Geral:

- Capacitar os alunos a compreender e aplicar os conceitos fundamentais relacionados à elaboração de algoritmos, estruturas de controle e estruturas de dados em linguagem de alto nível, visando a resolução de problemas computacionais de forma estruturada e eficiente.

2.2. Específicos:

- Compreender os elementos fundamentais de um algoritmo, como variáveis, operações primitivas, estruturas de controle e estruturas de dados.
- Aplicar comandos de entrada e saída para receber e exibir dados do usuário na tela ou em arquivos.
- Trabalhar com tipos de dados básicos, como inteiros, decimais, caracteres e booleanos.
- Realizar operações aritméticas, lógicas e relacionais utilizando operadores primitivos.
- Implementar estruturas de seleção (if, else, switch) para tomada de decisões com base em condições.
- Utilizar estruturas de repetição (for, while, do-while) para executar tarefas de forma repetitiva.
- Manipular estruturas de dados compostas, como vetores, matrizes e listas, para armazenar e organizar informações.
- Dividir o código em módulos ou funções para facilitar a compreensão, manutenção e reutilização.
- Documentar o código fonte de forma clara e concisa, utilizando comentários e descrições para explicar a lógica e o funcionamento do programa.

3. CONTEÚDO PROGRAMÁTICO									
U	**Conteúdos**	**Conhecimentos**	**Habilidades**	**Atitudes**	**Nº Horas /Aulas**				
					T	P	L	SP	EAD
I	**Introdução aos Algoritmos e Conceitos Fundamentais em Programação** - Introdução aos algoritmos e sua importância na resolução de problemas computacionais. - Tipos de algoritmos: descrição narrativa, fluxograma e pseudocódigo. - Estruturas condicionais em algoritmos: se, senão, escolha-caso. - Estruturas de repetição em algoritmos: para, enquanto, faca-enquanto. - Conceitos fundamentais em programação: edição, compilação, execução, depuração, programas e linguagens de programação. - Atribuição e entrada/saída de dados em algoritmos.	- Compreender o conceito de algoritmos e sua aplicação na resolução de problemas computacionais. - Identificar e descrever os tipos de algoritmos, incluindo descrição narrativa, fluxograma e pseudocódigo. - Reconhecer e aplicar estruturas condicionais em algoritmos, como os comandos "se", "senão" e "escolha-caso". - Familiarizar-se com estruturas de repetição em algoritmos, como os comandos "para", "enquanto" e "faça-enquanto". - Entender os conceitos fundamentais em programação, incluindo edição, compilação, execução, depuração, programas e linguagens de programação. - Saber como realizar atribuições e operações de entrada/saída de dados em algoritmos.	- Analisar e resolver problemas utilizando algoritmos. - Criar e representar algoritmos através de descrição narrativa, fluxograma e pseudocódigo. - Implementar estruturas condicionais ("se", "senão", "escolha-caso") em algoritmos. - Implementar estruturas de repetição ("para", "enquanto", "faça-enquanto") em algoritmos. - Utilizar ferramentas de desenvolvimento, como editores de código, compiladores e depuradores. - Manipular dados em algoritmos, realizando operações de atribuição e entrada/saída.	- Demonstrar curiosidade e interesse em aprender novos conceitos e técnicas de programação. - Valorizar a precisão e atenção aos detalhes na escrita de algoritmos e codificação de programas. - Mostrar persistência e resiliência na resolução de problemas e depuração de erros. - Colaborar e trabalhar em equipe em projetos e atividades em grupo. - Adotar uma postura ética e responsável no desenvolvimento e uso de algoritmos e programas.	10	10			
	Introdução a linguagem de programação Python. - Tipos de dados numéricos em Python.	- Compreender os tipos de dados numéricos em Python.	- Implementar e utilizar corretamente os tipos de dados numéricos em Python.	- Mostrar curiosidade e interesse em aprender e explorar a linguagem de programação Python.	15	15			

II	- Comandos de atribuição, leitura e exibição de dados e resultados. - Operações numéricas: aritméticas, lógicas e relacionais. - Precedência de operadores em expressões numéricas. - Conversão de tipos de dados numéricos em Python. - Operações booleanas: tipos booleanos e lógicos. - Estruturas de decisão em Python: if-else, elif, match-case. - Estruturas de repetição em Python: for e while. - Comandos condicionais e de repetição aninhados para iteração complexa.	Conhecer os comandos de atribuição, leitura e exibição de dados e resultados em Python. - Entender e aplicar operações numéricas: aritméticas, lógicas e relacionais em Python. - Reconhecer a precedência de operadores em expressões numéricas em Python. - Compreender a conversão de tipos de dados numéricos em Python. - Identificar e utilizar operações booleanas: tipos booleanos e lógicos em Python. - Entender e aplicar estruturas de decisão em Python: if-else, elif, match-case. - Familiarizar-se com estruturas de repetição em Python: for e while. - Compreender e implementar comandos condicionais e de repetição aninhados para iteração complexa em Python.	- Escrever e executar comandos de atribuição, leitura e exibição de dados e resultados em Python. - Realizar operações numéricas aritméticas, lógicas e relacionais em Python. - Aplicar corretamente a precedência de operadores em expressões numéricas em Python. - Realizar a conversão entre diferentes tipos de dados numéricos em Python. Implementar operações booleanas e utilizar tipos booleanos e lógicos em Python. - Escrever e implementar estruturas de decisão if-else, elif e match-case em Python. - Escrever e implementar estruturas de repetição for e while em Python. - Desenvolver e implementar comandos condicionais e de repetição aninhados para resolver problemas complexos em Python.	- Demonstrar precisão e atenção aos detalhes ao codificar em Python. - Persistir na resolução de problemas e depuração de códigos em Python. - Colaborar e trabalhar em equipe em projetos e atividades envolvendo Python. - Adotar uma postura ética e responsável ao desenvolver programas em Python.					
III	**Manipulação de Dados, Funções e Arquivos em Python** - Operações básicas para manipulação de strings: concatenação, fatiamento, busca e substituição. - Conceito de lista em Python e operações básicas: criação,	- Compreender operações básicas para manipulação de strings em Python: concatenação, fatiamento, busca e substituição. - Entender o conceito de lista em Python e as operações básicas: criação, inicialização, inserção, remoção e busca.	- Realizar operações básicas de manipulação de strings em Python, como concatenação, fatiamento, busca e substituição. - Criar, inicializar, inserir, remover e buscar elementos em listas em Python.	- Mostrar curiosidade e interesse em explorar as capacidades de manipulação de dados e criação de funções e arquivos em Python. - Demonstrar precisão e atenção aos detalhes na manipulação de strings, listas,	15	15			

	inicialização, inserção, remoção e busca. - Criação e importação de funções em Python - Manipulação de arquivos txt: criação, busca e remoção. - Criação de interface gráfica.	- Conhecer a criação e importação de funções em Python. - Compreender a manipulação de arquivos txt em Python: criação, busca e remoção. - Conhecer os conceitos básicos de criação de interface gráfica em Python.	- Escrever, criar e importar funções em Python para modularização e reutilização de código. - Manipular arquivos txt em Python, realizando operações de criação, busca e remoção. - Desenvolver interfaces gráficas básicas em Python.	funções e arquivos em Python. - Persistir na resolução de problemas relacionados à manipulação de dados, funções e arquivos em Python. - Colaborar e trabalhar em equipe em projetos que envolvem manipulação de dados e criação de interfaces gráficas em Python.					
Subtotal da Carga Horária					**40**	**40**			
Total da Carga Horária					**80**				

Legenda: U - unidade **T –** quantidade de aula(s) teórica(s) por unidade(s); **P-** quantidade de aula(s) práticas(s) por unidade(s); **L - quantidade de aula(s) por unidade(s) de Prática Pedagógica (exclusivo para o Curso de Educação Física - Licenciatura**); **SP** - quantidade de aulas(s) semipresenciais; **EAD** - quantidade de aula(s) à distância.

4. METODOLOGIA

O componente curricular será ministrado de forma dinâmica, agregando estratégias de metodologias ativas: aulas expositivas e dialogadas; exibição de vídeos; discussão de casos; experiências e situações do mercado de trabalho, através da metodologia de situações-problema; projetos de software; atividades práticas; atividade integrada (Prática Integradora), seguindo o modelo proposto para o período letivo do curso.

5. RECURSOS DIDÁTICOS

Projetor multimídia, vídeos, documentários, quadro branco e pincel que subsidiarão as aulas expositivo-dialogadas, os debates, as discussões e a atividade integrada. Para as sessões de filmes e vídeos, projetor de mídia e notebook. Para as atividades de estudo e pesquisa, livros didáticos disponíveis na biblioteca ou disponibilizados pelo docente, bem como artigos científicos disponíveis nas Bases de Dados (EBSCO e Portal CAPES). Ainda, será utilizado o Ambiente Virtual de Aprendizagem (AVA) do SECTRAS *On Line*.

6. AVALIAÇÃO (ÕES)

A avaliação do discente será contínua e processual. Se dará mediante o somatório de diferentes atividades, que constituirão uma nota final para cada uma das três principais verificações de aprendizagem (uma por UNIDADE) previstas pela Instituição. A média final será obtida a partir dos resultados de cada Unidade. Estão previstas 3 avaliações (formativas e somativas), dispostas da seguinte forma:

- 1ª Unidade: avaliação escrita (10,00).
- 2ª Unidade: avaliação escrita (8,00) e projeto de software (2,00).

- 3ª Unidade: avaliação escrita (6,00), projeto de software (2,00) e Prática Integradora (2,00).

7. BIBLIOGRAFIA

7.1. Básica:

ARAÚJO, S. **Linguagem de programação (ADS)**. 1. ed. São Paulo: Pearson Education do Brasil, 2020. [acervo digital]
ARAÚJO, S. **Lógica de programação e algoritmos.** 1. ed. São Paulo: Pearson Education do Brasil, 2020. [acervo digital]
CAETANO, M. A. L. **Python e mercado financeiro.** 1. ed. São Paulo: Blucher, 2021. [acervo digital]

7.2. **Complementar:**

ASCENCIO, AFG, ARAUJO, GS. **Estrutura de Dados: Algoritmos, análise da complexidade e implementações em Java e C/C++.** São Paulo: Pearson Education do Brasil, 2015. [acervo digital]
ASCENCIO, AFG, CAMPOS, EAV. **Fundamentos de Programação de Computadores.** São Paulo: Pearson Education do Brasil, 2012. [acervo digital]
GUEDES, Sérgio. **Lógica de Programação Algorítmica.** São Paulo: Pearson Education do Brasil, 2015. [acervo digital]
PUGA, Sandra; RISSETI, Gerson. **Lógica de programação e estrutura de dados, com aplicação em Java.** 3. ed. São Paulo: Pearson Education do Brasil, 2017. [acervo digital]
SIMÕES-PEREIRA, J. M. S. **Grafos e redes: teoria e algoritmos básicos.** 1. ed. Rio de Janeiro: Interciência, 2014. [acervo digital]