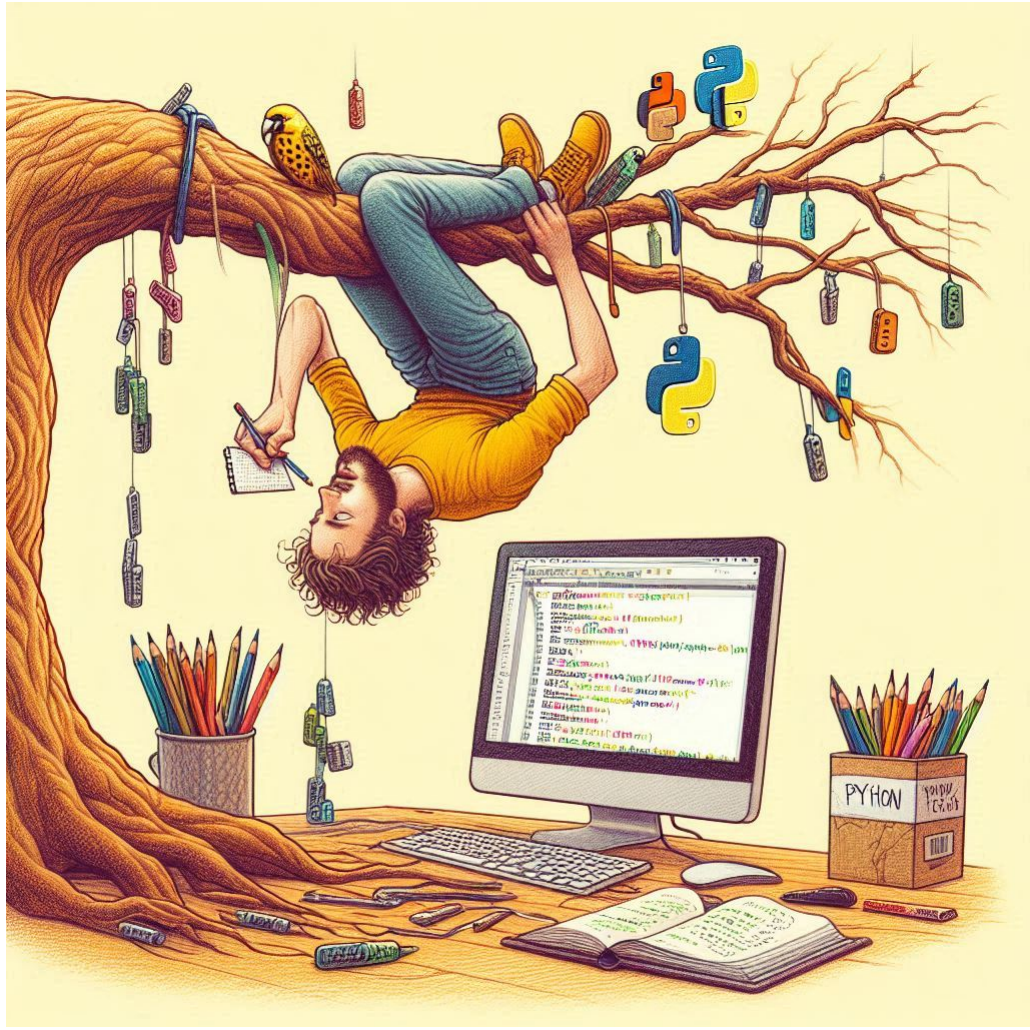
SI

SISTEMAS DE INFORMAÇÃO

Trabalho prático #1

Jornada na Floresta Trinária: Uma História de Conquista e Descoberta!

Em uma floresta misteriosa, Arthur, um jovem aventureiro, encontra uma árvore mágica que o leva a uma jornada épica. Através de três caminhos desafiadores, ele testa sua lógica, força e inteligência, desvendando segredos ancestrais e aprendendo a compreender a natureza em sua totalidade. Ao retornar à sua aldeia, Arthur compartilha seus conhecimentos e inspira a todos com sua bravura e perseverança. A história de Arthur nos ensina que, com determinação e busca pelo conhecimento, podemos superar qualquer obstáculo e alcançar grandes objetivos.



Objetivo:

O objetivo deste trabalho prático é implementar uma estrutura de dados em árvore ternária em Python. A árvore deve ter as seguintes propriedades:

- Cada nó possui três filhos: esquerdo, central e direito.
- O valor do filho esquerdo deve ser menor que o valor do pai.
- O valor do filho central deve ser igual ao valor do pai.
- O valor do filho direito deve ser maior que o valor do pai.

Requisitos:

- Implementar a classe Node para representar um nó na árvore. A classe deve ter os seguintes atributos:
  - value: O valor do nó.
  - left: Uma referência ao filho esquerdo do nó.
  - middle: Uma referência ao filho central do nó.
  - right: Uma referência ao filho direito do nó.

Requisitos (continuação):

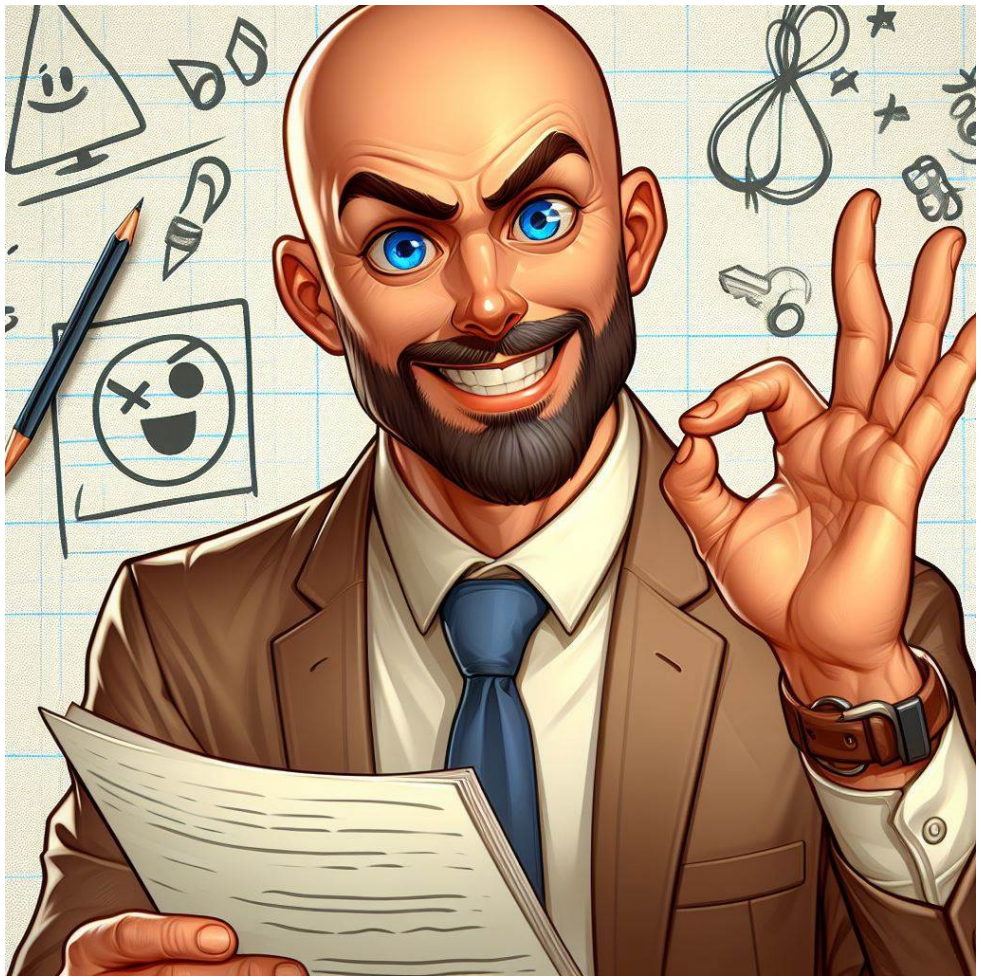
- Implementar a classe TernaryTree com as seguintes operações na árvore:
  - insert(value): Insere um novo valor na árvore.
  - search(value): Busca um valor na árvore e retorna o nó correspondente, se encontrado.
  - delete(value): Exclui um valor da árvore.
  - inorder(): Percorre a árvore em ordem e imprime os valores dos nós.
  - preorder(): Percorre a árvore em pré-ordem e imprime os valores dos nós.
  - postorder(): Percorre a árvore em pós-ordem e imprime os valores dos nós.

Entregas

- Trabalho escrito, em formato de relatório técnico
- Teoria básica de apoio à implementação
- Diagrama de blocos do software desenvolvido
- Explicação funcional de cada bloco de software
- Arquivo .py com o código funcional

Avaliação

- Verificação de conteúdo gerado por IA
- Organização e apresentação do trabalho escrito
- Uso das estruturas de dados discutidas em aula
- Explicação dos blocos funcionais do código



Trabalho prático #2

A Biblioteca Balanceada!

A Biblioteca Municipal, antes desorganizada, encontrou na Árvore AVL a solução para gerenciar seu acervo de forma eficiente, permitindo aos leitores encontrarem seus livros favoritos com rapidez e precisão. A estrutura inteligente garantiu organização, agilidade e acesso rápido à literatura, transformando a biblioteca em um paraíso para os amantes da leitura.

Objetivo:

Implementar uma árvore AVL em Python, com todas as operações e requisitos necessários para gerenciar um acervo de forma eficiente.

Requisitos:

- Definição da Classe Árvore AVL:
  - Criar uma classe AVLTree para representar a árvore AVL.
  - Implementar métodos para inicializar a árvore, inserir, remover e buscar elementos na árvore.
  - Implementar métodos para calcular o fator de balanceamento e realizar rotações simples (à direita e à esquerda) e duplas (à direita e à esquerda).
  - Implementar métodos para verificar se a árvore está balanceada e para imprimir a árvore em diferentes formatos (pré-ordem, ordem e pós-ordem).

Requisitos:

- Operações Básicas:
  - Inserção: Inserir elementos na árvore AVL, mantendo o balanceamento da árvore após cada inserção.
  - Remoção: Remover elementos da árvore AVL, atualizando os ponteiros e fatores de balanceamento dos nós envolvidos.
  - Busca: Buscar um elemento específico na árvore AVL, percorrendo a árvore de forma recursiva.

GOLDEN FEATURE (+1,0)

- Gerar uma representação GRÁFICA da árvore ternária.

Entregas

- Trabalho escrito, em formato de relatório técnico
- Teoria de base
- Diagrama de blocos do software desenvolvido
- Explicação funcional de cada bloco de software
- Arquivo .py com o código funcional

Avaliação

- Verificação de conteúdo gerado por IA
- Organização e apresentação do trabalho escrito
- Uso das estruturas de dados discutidas em aula
- Explicação dos blocos funcionais do código