

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
ESCOLA POLITÉCNICA UFRJ

GABRIEL SANTA BARBARA RODRIGUES SOUZA - 123356007

JULIA FERNANDA TERRA SOUZA - 123460458

SAMUEL DO NASCIMENTO CAROLA - 123537899

ABB e índices de SBD

Rio de Janeiro

2024

## Comentários sobre o Desenvolvimento

### Estruturas de Dados Utilizadas

- **Estrutura Linear (EDL):** Lista Python que armazena os registros do banco de dados (cada registro contém CPF, nome, data de nascimento e flag de deletado).
- **Árvore Binária de Busca (ABB):** Utilizada como índice para os registros, permitindo operações eficientes de busca, inserção e remoção baseadas no CPF.
- **Classes Principais:**
  - Registro: Representa cada pessoa com CPF, nome, data de nascimento e flag de deletado.
  - NoABB: Nó da árvore de busca, guarda a chave (CPF), índice do registro e ponteiros para os filhos.
  - ABB: Implementa as operações da árvore de busca (inserção, remoção, busca, percursos).
  - SGBD: Gerencia a EDL, o índice ABB e oferece as operações de alto nível para o usuário (inserir, remover, buscar, listar, persistir).

### Divisão de Módulos

- **ABB.py:** Contém as implementações das estruturas de dados, lógica da árvore, e das operações do SGBD.
- **interface.py:** Implementa a interface gráfica usando Tkinter, conectando as operações do SGBD ao usuário final. (apenas para uma melhor interação), mas o código tem exemplos de uso sem interface e funciona atendendo aos problemas pedidos.

### Descrição das Rotinas e Funções

- **Inserção:** Verifica duplicidade do CPF, adiciona registro à EDL e insere na ABB.
- **Remoção:** Marca o registro como deletado na EDL e remove o CPF do índice ABB.
- **Busca:** Utiliza o índice ABB para encontrar rapidamente o registro pelo CPF.
- **Listagem Ordenada:** Realiza percurso em ordem na ABB para listar os registros não deletados em ordem crescente de CPF.
- **Persistência:** Salva e carrega os registros da EDL em arquivo binário usando o módulo pickle.
- **Interface:** Permite inserir, buscar, remover e listar registros por meio de uma aplicação gráfica simples.

### Complexidades de Tempo e Espaço

- **Busca, Inserção e Remoção na ABB:**  $O(\log n)$  no caso médio (árvore balanceada); pior caso  $O(n)$  se a árvore ficar desbalanceada.
- **Listagem Ordenada:**  $O(n)$ , pois percorre todos os nós da árvore em ordem.

- **Persistência:**  $O(n)$  para salvar/carregar todos os registros.
- **Espaço:** Linear em relação ao número de registros (armazenamento duplicado: EDL e ABB como índice).

### Problemas e Observações Encontrados

- **Desbalanceamento da ABB:** A árvore pode ficar desbalanceada em certas ordens de inserção/remoção, degradando o desempenho das operações para  $O(n)$ .
- **Remoção Lógica:** Os registros não são removidos fisicamente da EDL, apenas marcados como deletados, o que pode aumentar o uso de memória ao longo do tempo.
- **Persistência Simples:** O uso de pickle é suficiente para projetos pequenos, mas não é recomendado para grandes volumes ou produção.
- **Validação Básica:** A interface e as rotinas validam apenas o formato do CPF e campos obrigatórios; não há validação mais profunda dos dados.
- **Interface Gráfica Simples:** O foco do projeto está mais na lógica do SGBD do que na interface do usuário.

### Conclusão sobre os Resultados Obtidos

O projeto atingiu seu objetivo de demonstrar um SGBD simples com indexação por árvore binária de busca, possibilitando inserção, busca e remoção eficientes de registros pelo CPF. A utilização da ABB como índice proporciona desempenho adequado para bases pequenas e médias, e a separação dos módulos facilita a manutenção e possíveis expansões. Entretanto, a ausência de balanceamento da árvore e a remoção apenas lógica apontam limitações para cenários de uso mais intensivo. Para aplicações maiores ou produção, recomendam-se melhorias como balanceamento automático da ABB (AVL ou Red-Black Tree), remoção física de registros deletados e estratégias de persistência mais robustas.

Em resumo, o projeto é didático e cumpre seu papel como exemplo de uso de ABB em SGBDs, com código bem estruturado e comentado, evidenciando decisões, limitações e potencial de evolução.