Relatório do Processo de Criação do Trabalho: Tabela Hash para Deduplicação de Dados

1. Objetivo do Trabalho O projeto consistiu em criar uma estrutura de dados do tipo Tabela Hash para identificar e remover duplicatas em um conjunto de dados, utilizando Python e a biblioteca Pandas. O foco principal foi trabalhar com dados em formato CSV contendo CPFs como chave única para a deduplicação.
2. Desenvolvimento Inicial e Primeiros Códigos Foi desenvolvido inicialmente um código básico em Python que implementava a Tabela Hash com métodos para inserir, buscar, remover e listar dados.

O código permitia a leitura de um arquivo CSV e inserção dos dados na tabela para eliminar duplicatas com base na chave CPF.

O script incluía upload e download de arquivos para facilitar o uso no Google Colab.

3. Erros Encontrados e Correções Realizadas Erro 1: KeyError: 'CPF' O código lançava um erro indicando que a coluna 'CPF' não existia no DataFrame.

Causa: O arquivo CSV usava ponto e vírgula (;) como separador, padrão comum em arquivos exportados do Excel em português, e não vírgula.

Correção: Alteração do parâmetro sep=';' no pd.read_csv() para carregar corretamente as colunas.

Erro 2: Espaços extras nos nomes das colunas As colunas tinham espaços em branco, por exemplo ' CPF' ou 'nome '.

Isso causava dificuldades na referência das colunas no código.

Correção: Uso do comando df.columns = df.columns.str.strip() para remover espaços extras.

Erro 3: Distribuição ruim na tabela hash (baldes vazios) Observou-se que a função hash usada (hash_divisao simples) mapeava as chaves diretamente pelo resto da divisão por 1000, resultando em muitos baldes vazios e poucos baldes com dados.

Isso impactava a eficiência e a organização da tabela.

Correção: Implementação de uma função de hash mais sofisticada pelo método da multiplicação (usando o número áureo) e ajuste do tamanho da tabela para um número primo (997) para melhor distribuição.

4. Quantidade de Códigos Desenvolvidos Durante o processo, foram criadas aproximadamente 3 versões principais do código:

Versão inicial básica — com TabelaHash simples, upload/download e hash por divisão básica.

Versão corrigida para leitura correta do CSV — com ajuste do separador, strip dos nomes das colunas, para resolver o KeyError.

Versão otimizada da Tabela Hash — com função de hash melhorada (multiplicação), tamanho da tabela primo e melhorias na distribuição dos dados.

Cada versão teve adaptações para execução no Google Colab, com a inclusão do upload automático do arquivo CSV e do download do arquivo final sem duplicatas.

5. Aprendizados e Considerações Finais A importância de conhecer o formato real dos dados (por exemplo, separadores no CSV) para evitar erros comuns na importação.

Apos realizar os testes foi notada a replicação de todos os espaços vazios ou numero portanto foi necesarria uma correção no codigo . O que mudou? Adicionei o print do DataFrame original com todos os dados carregados, antes da inserção na tabela hash.

- Criei o método baldes_com_dados para listar só os baldes que contém elementos e evitar imprimir listas vazias.
- Modifiquei o método **str** para imprimir só os baldes com dados.
- Imprimi a tabela hash logo após inserir os dados para ver como ficaram os baldes sem mostrar os vazios.
- Imprimi o DataFrame resultante com os dados únicos após a deduplicação e salvei em um novo arquivo para verificar se estava correto.

Davidson Diógenes Vasconcelos da Silva Santos

DRE: 123531495

Mônica de Sousa Amaral

DRE: 119160444

Naya da Silva Nascimento

DRE: 119160428