Centro Universitário de Brasília (CEUB)

Faculdade de Tecnologia e Ciências Sociais Aplicadas (FATECS)

Thales Rassi Porto de Matos - 22400186

Gabriel Marques da Rocha - 22451254

Gabrielle Gutierres - 22350026

Pedro Klein - 22105154

Matheus de Morais - 22352763

Henrique Lessa - 22402204

Documentação ETL – POPULAÇÃO DF PLANO DE SAUDE

Brasília

2025

Thales Rassi Porto de Matos

Gabriel Marques da Rocha

Gabrielle Gutierres

Pedro Klein

Matheus de Morais

Henrique Lessa

Documentação ETL – POPULAÇÃO DF PLANO DE SAUDE

Atividade final apresentada à Faculdade de Tecnologia e Ciências Sociais Aplicadas (FATECS)
, do Centro Universitário de Brasília (CEUB) como parte integrante do currículo da disciplina Interação Humano Computador, da graduação em Ciência da computação
Professora responsável: Kadidja Valeria Reginaldo de Oliveira

Brasília

2025

SUMÁRIO

IDENTIFICAÇÃO

Script: ETL_POPULACAO_DF_PLANO_DE_SAUDE.py
Responsável(is): Equipe Radiologia DF

OBJETIVO DO ETL

Extrair do dataset oficial a população total e a distribuição entre população com e sem plano de saúde por Região Administrativa (RA) do Distrito Federal, padronizar e higienizar os dados, mapear as RAs para seus respectivos identificadores no banco e inserir esses registros na tabela populacao.

Este ETL garante que a base populacional do projeto Radiologia DF esteja normalizada, atualizada e integrada com a dimensão de Regiões Administrativas.

FONTES DE DADOS DE ENTRADA

Arquivo principal (dataset de população por RA):

1. dirty_data_pop_df_com_plano_saude_por_ra.csv
    - Caminho esperado: diretório local do projeto
    - Formato: CSV
    - Separador: vírgula (";")
    - Encoding: padrão do pandas (utf-8, salvo ajuste no arquivo)
    - Periodicidade de atualização: conforme disponibilização dos dados populacionais oficiais por RA (anual)

Observação:

- O ano associado à população é definido pela constante ANO_REFERENCIA = 2021.

TABELAS DE DESTINO IMPACTADAS

1. populacao
    - Tabela que armazena a população total, com plano de saúde e sem plano de saúde para cada RA e ano.

FLUXO RESUMIDO DO ETL

Passo 1 – Tratamento do dataset bruto

- Verifica se o arquivo definido em DATASET_PATH existe; caso contrário, lança FileNotFoundError.
- Lê o CSV com separador ";" e engine "python".
- Imprime as colunas encontradas para conferência.
- Valida a existência das colunas obrigatórias:
    - "Local"
    - "Total"
    - "Sim"
    - "Nao"
- Constrói um DataFrame tratado (df_out) contendo:
    - nome_ra → obtido a partir de Local, usando a função limpar_nome_ra (strip, title case, validação de vazio)
    - populacao_total → convertida para número, substituindo valores inválidos por 0, convertida para inteiro e limitada a valores ≥ 0
    - populacao_com_plano_saude → mesma lógica da coluna anterior
    - populacao_sem_plano_saude → mesma lógica anterior
    - ano → valor fixo definido pela constante ANO_REFERENCIA = 2021
- Remove linhas onde nome_ra é nulo.

Passo 2 – Mapeamento de id_ra

- Consulta a tabela regiao_administrativa: SELECT id_ra, nome FROM regiao_administrativa;
- Cria um dicionário mapeando nome (lowercase e strip) → id_ra.
- Atribui ao DataFrame a coluna **id_ra**, usando esse dicionário.
- Identifica e imprime as RAs não encontradas no banco.

- Remove linhas cujo id_ra seja nulo, garantindo que apenas RAs válidas serão carregadas.

Passo 3 – Inserção na tabela populacao

- Monta uma lista de tuplas com os valores: (id_ra, ano, populacao_total, populacao_com_plano_saude, populacao_sem_plano_saude)
- Se a lista estiver vazia, imprime aviso e encerra.
- Caso haja registros:
- Insere todos em lote via execute_values:
    - INSERT INTO populacao (id_ra, ano, populacao_total, populacao_plano_saude, populacao_sem_plano_saude) VALUES %s;
- Realiza commit.
- Imprime o total de registros inseridos.

Passo 4 – Encerramento

- Fecha cursor e conexão com o banco.
- Imprime mensagem final indicando a conclusão do ETL.

MAPEAMENTO DE CAMPOS (ORIGEM → DESTINO):

"Local" → nome_ra → regiao_administrativa.id_ra (após mapeamento)

"Total" → populacao_total

"Sim" → populacao_com_plano_saude

"Nao" → populacao_sem_plano_saude

ANO_REFERENCIA (2021) → ano

DEPENDÊNCIAS E PRÉ-REQUISITOS

Bibliotecas Python:

- os
- pandas
- psycopg2
- psycopg2.extras
- python-dotenv (load_dotenv)
- config_db (função get_conn para conexão com o banco)

Tabelas:

- regiao_administrativa (deve conter todos os nomes de RA exatamente como padronizados pelo script)
- populacao (schema compatível com os campos inseridos)

Requisitos de dados

- Arquivo CSV existente no caminho definido em DATASET_PATH
- Colunas "Local", "Total", "Sim", "Nao" presentes
- Dados numéricos válidos ou coerentes nas colunas populacionais

COMO EXECUTAR

- Garantir que o ambiente virtual (se houver) esteja ativado.
- Garantir que o arquivo .env esteja configurado com os parâmetros de conexão ao banco.
- Verificar que o arquivo dirty_data_pop_df_com_plano_saude_por_ra.csv está no caminho esperado.
- Executar: python3 ETL_POPULACAO_POR_RA.py

Pré-condições:

- Banco PostgreSQL acessível.
- Tabela regiao_administrativa populada e padronizada.
- Tabela populacao criada.