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 – TIPO EXAME

Brasília

2025

Thales Rassi Porto de Matos

Gabriel Marques da Rocha

Gabrielle Gutierres

Pedro Klein

Matheus de Morais

Henrique Lessa

Documentação ETL – TIPO EXAME

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_TIPO_EXAME.py
Responsável(is): Equipe Radiologia DF

OBJETIVO DO ETL

Extrair automaticamente os nomes dos tipos de exames de diagnóstico por imagem a partir das colunas numéricas do dataset histórico de exames mais requisitados, padronizar esses nomes, complementar a lista com tipos de exame fixos definidos pelo projeto e inserir novos tipos na tabela tipo_exame, sem duplicidades.

Este ETL garante que a dimensão de tipo de exame esteja sempre completa, atualizada e alinhada aos datasets de origem.

FONTES DE DADOS DE ENTRADA

Arquivo principal:

1. dirty_data_historico_grupos_exames_img_mais_requisitados.csv
    - Caminho esperado: diretório local do projeto
    - Formato: CSV
    - Separador: “;”
    - Encoding: padrão do pandas
    - Periodicidade: conforme novos dados históricos de exames

TABELAS DE DESTINO IMPACTADAS

1. tipo_exame
    - Tabela contendo os nomes dos tipos de exame
    - Campos afetados:
        - nome
        - descricao (armazenada como NULL)
        - Este ETL insere apenas tipos novos, sem atualizar registros existentes.

FLUXO RESUMIDO DO ETL

Passo 1 – Extração dos nomes de exame

- Lê o CSV definido em DATASET_EXAMES com separador ";".
- Identifica como tipos de exame todas as colunas que:
    - não estão em COLUNAS_IGNORAR
    - possuem dtype numérico (df[col].dtype != object)
- Imprime as colunas detectadas no console.
- Cada nome bruto de coluna passa pela função limpar_nome_exame, que aplica:
    - strip de espaços
    - conversão para minúsculas
    - remoção de números
    - troca de _ e - por espaço
    - remoção de conteúdo entre parênteses
    - redução de múltiplos espaços
    - conversão para Title Case
    - Após limpar os nomes:
        - adiciona também os TIPOS_EXAME_FIXOS
        - remove duplicidades
        - ordena alfabeticamente
        - retorna um DataFrame com a coluna: nome_exame

Passo 2 – Inserção na tabela tipo_exame

- Consulta todos os nomes já existentes na tabela tipo_exame.
- Compara cada nome extraído com os existentes.
- Monta lista contendo apenas tipos novos: (nome_exame, descricao=None)
- Caso haja novos tipos:
    - Insere em lote via execute_values: INSERT INTO tipo_exame (nome, descricao) VALUES %s;
    - Realiza commit
    - Imprime quantos tipos foram inseridos
- Caso contrário:
    - Imprime “Nenhum tipo novo para inserir.”

Passo 3 – Encerramento

- Fecha o cursor
- Fecha a conexão com o banco
- Finaliza o processo

MAPEAMENTO DE CAMPOS (ORIGEM → DESTINO):

Origem no CSV	Destino
Nome da coluna numérica do CSV	tipo_exame.nome (após normalização)
TIPOS_EXAME_FIXOS	tipo_exame.nome
descricao	sempre NULL

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:

- tipo_exame (já criada)

Requisitos de dados

- Arquivo CSV existente no caminho definido
- Colunas numéricas representando tipos de exame
  Colunas ignoradas presentes conforme especificado

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.
- Garantir que o CSV está no caminho definido em DATASET_EXAMES.
- Executar: python3 ETL_TIPO_EXAME.py

Pré-condições:

- Banco PostgreSQL acessível
- Tabela tipo_exame criada
- Dataset limpo e compatível