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 – WAVE

Brasília

2025

Thales Rassi Porto de Matos

Gabriel Marques da Rocha

Gabrielle Gutierres

Pedro Klein

Matheus de Morais

Henrique Lessa

Documentação ETL – WAVE

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_WAVE.py
Responsável(is): Equipe Radiologia DF

OBJETIVO DO ETL

Processar automaticamente o dataset gerado pela ferramenta WAVE contendo métricas de acessibilidade e usabilidade de páginas oficiais de sistemas públicos (DATASUS, CNES, IBGE, SISCAN, IPEDF), identificar o sistema responsável por cada URL, higienizar e normalizar as métricas coletadas, registrar novas páginas na pagina_portal e inserir os indicadores de acessibilidade na tabela metrica_wave.

Este ETL garante rastreabilidade, consistência temporal e completude das métricas de usabilidade que alimentam análises de qualidade e acessibilidade dos portais utilizados no projeto Radiologia DF.

FONTES DE DADOS DE ENTRADA

Arquivo principal:

1. dirty_data_dataset_usabilidade_dataSUS_WAVE.csv
    - Caminho esperado: diretório local do projeto
    - Formato: CSV
    - Separador: padrão do pandas
    - skiprows: 2 (as duas primeiras linhas são descartadas pois não fazem parte da tabela)
    - Encoding: padrão do pandas
    - Periodicidade: sempre que uma nova coleta automática WAVE for realizada

TABELAS DE DESTINO IMPACTADAS

1. pagina_portal
    - A tabela armazena informações sobre as páginas avaliadas.
2. metrica_wave
    - A tabela armazena registros de cada coleta WAVE para cada páginal

FLUXO RESUMIDO DO ETL

Passo 1 – Tratamento do dataset bruto

- Carrega o CSV ignorando as duas primeiras linhas (skiprows=2).
- Remove espaços das colunas.
- Renomeia colunas conforme mapeamento do script.
- Seleciona somente as colunas necessárias:
    - url
    - errors
    - contrast_errors
    - alerts
    - aim_score
- Normaliza a coluna url com strip.
- Determina o campo sistema, aplicando classificar_sistema() baseado na URL:
    - "?cnes" → CNES
    - "?ibge" → IBGE
    - "?pnad" → IPEDF
    - "?siscan" → SISCAN
    - caso contrário → DATASUS
- Normalização de métricas:
    - Converte errors, contrast_errors e alerts para valores inteiros:
    - valores inválidos são substituídos por 0
    - negativos são forçados para 0
    - arredonda para baixo com floor()
    - Converte aim_score para float (valores inválidos viram NaN)
- Resultado: DataFrame final contendo as colunas:
    - url
    - sistema
    - errors
    - contrast_errors
    - alerts
    - aim_score

Passo 2 – Carregamento em pagina_portal

- Busca URLs já existentes no banco: SELECT id_pagina, url FROM pagina_portal;
- Identifica URLs novas no dataset.
- Para cada nova URL, insere:
    - nome = url

    - url = url
    - sistema = sistema detectado
- Inserção em lote: INSERT INTO pagina_portal (nome, url, sistema) VALUES %s
- É impresso no console o número de novas páginas inseridas.

Passo 3 – Carregamento em metrica_wave

- Busca novamente todas as páginas existentes: SELECT id_pagina, url FROM pagina_portal;
- Mapeia url → id_pagina
- Para cada linha do dataset tratado:
    - Obtém id_pagina
    - Se não existir, registra aviso e ignora a métrica
    - Se existir, constrói o registro:
        - (id_pagina, DATA_COLETA_FIXA, errors, contrast_errors, alerts, aim_score)
- Insere todas as métricas válidas em lote: INSERT INTO metrica_wave (id_pagina, data_coleta, errors, contrast_errors, alerts, aim_score) VALUES %s
- Realiza commit
- Imprime quantos registros foram inseridos

Passo 4 – Encerramento

- Fecha o cursor
- Fecha a conexão com o banco
- Imprime mensagem final confirmando que o ETL foi concluído

MAPEAMENTO DE CAMPOS (ORIGEM → DESTINO):

Origem no CSV	Destino no Banco
url	pagina_portal.url e pagina_portal.nome
sistema (derivado da URL)	pagina_portal.sistema

errors	metrica_wave.errors
contrast_errors	metrica_wave.contrast_errors
alerts	metrica_wave.alerts
aim_score	metrica_wave.aim_score
DATA_COLETA_FIXA	metrica_wave.data_coleta
url → id_pagina	FK em metrica_wave

DEPENDÊNCIAS E PRÉ-REQUISITOS

Bibliotecas Python:

- os
- pandas
- numpy
- psycopg2
- psycopg2.extras
- python-dotenv
- config_db (get_conn)

Tabelas:

- pagina_portal
- metrica_wave

Requisitos dos dados:

- CSV existente no caminho definido em DATASET_PATH
- Colunas esperadas presentes
- URLs devem ser únicas para cadastro correto em pagina_portal

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_dataset_usabilidade_dataSUS_WAVE.csv** está no caminho definido..
- Executar: python3 ETL_WAVE.py

Pré-condições:

- Banco PostgreSQL acessível
- Tabelas pagina_portal e metrica_wave existentes
- Arquivo CSV com estrutura compatível