

Sidra PI

Este banco de dados oferece um acesso facilitado e segmentado às principais tabelas do SIDRA, especificamente adaptadas para o estado do Piauí, permitindo análises detalhadas e personalizadas dos dados socioeconômicos e demográficos da região.

Funcionalidade geral dos scripts

sidrapi.py

Importações e Configuração Inicial:

- Bibliotecas Utilizadas: BeautifulSoup, pandas, numpy, time, locale, requests.

O código fornecido é dividido em duas classes principais: SidraManager e old_sidra_manager.

Inicialização: Define o código UF para Piauí (uf_code=22) e inicializa estruturas para gerenciamento de dados e erros.

A classe SidraManager oferece funcionalidades para interagir com a API do SIDRA, obtendo metadados de tabelas, processando esses metadados em DataFrames e buscando dados de tabelas específicas. Métodos principais incluem:

- sidra_get_metadata(número_tabela): Obtém os metadados de uma tabela específica.
- retry_failed_requests(delay_seconds): Tenta novamente buscar metadados para tabelas que falharam na primeira tentativa.
- sidra_process_table(dados): Processa os metadados de uma tabela em dois DataFrames.
- sidra_process_variables(dados, request_id): Processa os dados das variáveis de uma tabela.
- sidra_process_categories(dados, request_id): Processa os dados das categorias de uma tabela.
- fetch_sidra_data(t, v, c, n, p, f, d, h): Busca dados de uma tabela do SIDRA e retorna um DataFrame.

A classe old_sidra_manager oferece funcionalidades semelhantes para a versão antiga do SIDRA. Seus métodos principais são:

- sidra_info(número_tabela): Obtém informações sobre uma tabela específica.
- get_class_code(html_content): Extrai o código de classificação de uma tabela.
- get_territorial_code(html_content): Extrai o código territorial de uma tabela.
- sidra_get_vars(html_content): Obtém informações sobre as variáveis de uma tabela.

Ambas as classes fornecem uma interface para interagir com os dados do SIDRA, permitindo a obtenção e processamento eficientes de informações estatísticas do IBGE.

sidra-airflow.py

- task_get_data é responsável por obter os dados do SIDRA utilizando as informações previamente configuradas
- task_configure_paths: Esta função configura os caminhos necessários e salva informações importantes no XCom, mas não executa nenhuma operação crítica de obtenção de dados.
- task_configure_managers: Esta função configura os gerenciadores necessários para lidar com os diferentes aspectos do processo, mas também não executa a obtenção de dados.

- `task_get_data_info`: Esta função obtém as informações necessárias para a obtenção dos dados do SIDRA, mas ainda não realiza a obtenção efetiva dos dados.
- `task_get_data`: Esta função efetua a obtenção real dos dados do SIDRA com base nas informações obtidas anteriormente. É a parte crucial do processo de obtenção de dados.
- `task_configure_drive_repository`: Esta função configura o repositório de dados no Google Drive com base nas informações e dados obtidos anteriormente. Embora seja uma etapa importante para armazenar os dados, depende diretamente dos dados já obtidos.

operadores.py

Possui duas classes principais: `GoogleDriveManager` e `GoogleSheetManager`, que são responsáveis por gerenciar arquivos e planilhas no Google Drive e no Google Sheets, respectivamente.

`GoogleDriveManager`:

- **init**: Inicializa a classe com os caminhos para as credenciais e informações necessárias.
- `__authenticate`: Autentica a instância para uso dos serviços do Google Drive e Google Sheets.
- `get_file_info_by_url`: Obtém informações de um arquivo ou pasta a partir de sua URL.
- `create_folder`: Cria uma nova pasta no Google Drive.
- `create_spreadsheet`: Cria uma nova planilha no Google Sheets.
- `move_spreadsheet_to_folder`: Move uma planilha para uma pasta específica no Google Drive.
- `share_with_user`: Compartilha um arquivo ou pasta com um usuário específico.
- `delete_folder`: Exclui uma pasta e todo o seu conteúdo no Google Drive.
- `delete_file`: Exclui um arquivo específico no Google Drive.
- `list_all_contents`: Lista todos os conteúdos de uma pasta específica ou do diretório raiz.

`GoogleSheetManager`:

- **init**: Inicializa a classe com o caminho para as credenciais do Google Sheets.
- `get_data_info_from_url`: Obtém dados de uma planilha do Google Sheets como um `DataFrame` do `pandas`.
- `insert_data`: Insere dados em uma posição específica de uma página do Google Sheets.
- `insert_data_from_df`: Insere dados a partir de `pandas DataFrames`.
- `create_new_sheet`: Cria uma nova página (worksheet) em uma planilha do Google Sheets.
- `insert_rows`: Insere linhas em uma página existente do Google Sheets.
- `add_worksheet`: Adiciona uma nova página (worksheet) à planilha do Google Sheets.
- `delete_worksheet`: Deleta uma página específica da planilha do Google Sheets.
- `list_worksheets`: Lista todas as páginas (worksheets) de uma planilha do Google Sheets.

estruturas.py

A classe `InformationManager`, foi projetada para gerenciar informações armazenadas em um arquivo JSON.

- **init**: Inicializa a classe com o caminho do arquivo JSON e define o momento da última atualização como o momento de inicialização.
- `create_json`: Cria um novo arquivo JSON com informações básicas, incluindo administradores, base de conhecimento, última atualização e uma pasta principal vazia.
- `add_subfolders_to_folder`: Adiciona subpastas a uma pasta específica no arquivo JSON.
- `add_spreadsheets_to_subfolder`: Adiciona planilhas a uma subpasta específica no arquivo JSON.
- `add_sheets_to_spreadsheet`: Adiciona folhas a uma planilha específica no arquivo JSON.

- `get_folders_or_subfolders`: Obtém uma lista de todas as subpastas de uma pasta específica no arquivo JSON.
- `get_spreadsheets`: Obtém uma lista de todas as planilhas de uma subpasta específica no arquivo JSON.
- `get_spreadsheets_from_folder`: Obtém uma lista de todas as planilhas de todas as subpastas de uma pasta específica no arquivo JSON.
- `get_sheets_from_spreadsheet`: Obtém uma lista de todas as folhas de uma planilha específica no arquivo JSON.
- `update_field_by_path`: Atualiza um campo específico no arquivo JSON, fornecendo o caminho até o campo e o novo valor a ser atribuído.

configure_directory.py

A classe `DirectoryManager`, facilita o gerenciamento de diretórios e arquivos.

- **init**: Inicializa a classe com os diretórios de origem e destino.
- `_list_files`: Lista todos os arquivos no diretório de origem e retorna um `DataFrame` com os nomes dos arquivos.
- `_organize_files`: Organiza os arquivos listados no `DataFrame` em subpastas no diretório de destino com base nas pastas e subpastas especificadas no `DataFrame` de entrada.
- `execute_organize_files`: Executa a organização de arquivos usando as informações de pastas e subpastas fornecidas em um `DataFrame`.
- `process_template`: Processa um modelo Excel e preenche uma nova planilha com dados de um `DataFrame`, mantendo a formatação do modelo.

configure_dag.py

Essas funções em conjunto fornecem uma estrutura para automatizar a obtenção, organização e armazenamento de dados provenientes da API SIDRA do IBGE em planilhas do Google Sheets, mantendo a estrutura e os metadados das informações.

- `configure_paths`: Configura os caminhos para os arquivos de credenciais, informações e base de conhecimento.
- `configure_managers`: Configura os gerenciadores para interagir com os serviços e dados, incluindo o gerenciador de planilhas do Google, o gerenciador de unidades do Google Drive, o gerenciador de informações (JSON) e o gerenciador da API SIDRA.
- `get_sidra_api_info`: Obtém informações detalhadas dos dados disponíveis na API SIDRA do IBGE, como tabelas, variáveis e grupos, e as salva em um arquivo Excel.
- `configure_initial_sheet`: Configura uma planilha inicial com informações de uma tabela específica da API SIDRA para ser usada como modelo para preencher as outras planilhas.
- `configure_drive_repository`: Configura um repositório no Google Drive, criando uma pasta principal e subpastas com base nas informações disponíveis. Também compartilha a pasta principal com os administradores.
- `read_and_populate_spreadsheets`: Lê a lista de planilhas do arquivo JSON de informações e popula as planilhas com os dados da API SIDRA correspondentes.

init.py

Centraliza as importações necessárias para facilitar o acesso às classes e funções relevantes dentro do pacote `dags`.

- InformationManager: Importa a classe InformationManager do módulo estruturas, que provavelmente é responsável por gerenciar informações em formato JSON.
- GoogleSheetManager e GoogleDriveManager: Importam as classes GoogleSheetManager e GoogleDriveManager do módulo operadores, que provavelmente lidam com operações relacionadas a planilhas e arquivos no Google Drive, respectivamente.
- SidraManager: Importa a classe SidraManager do módulo sidrapi, que provavelmente é responsável por interagir com a API SIDRA do IBGE para obter dados estatísticos.
- DirectoryManager: Importa a classe DirectoryManager do módulo configure_directory, que provavelmente lida com a organização de diretórios e arquivos em um sistema de arquivos local.

configure_paths, configure_managers, get_sidra_api_info, get_sidra_api_data e configure_drive_repository: Importa várias funções do módulo configure_dag, que provavelmente são responsáveis por configurar caminhos, gerenciadores, obter informações da API SIDRA, processar dados da API SIDRA e configurar o repositório no Google Drive, respectivamente.

Funcionalidade geral do painel

O Painel Power BI foi desenvolvido para fornecer acesso fácil e intuitivo aos dados do banco de dados, permitindo que os usuários explorem e visualizem informações importantes de maneira eficiente. Este documento destina-se a fornecer uma visão geral das funcionalidades e recursos oferecidos pelo painel.

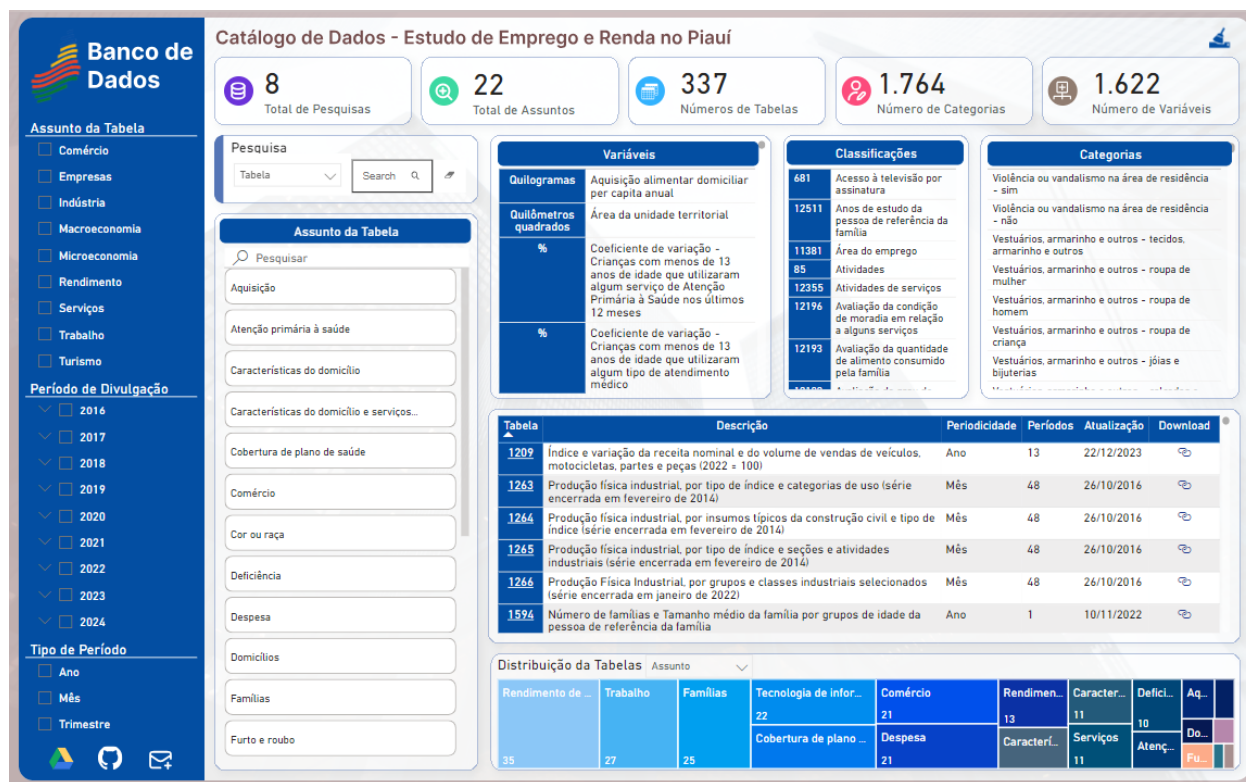
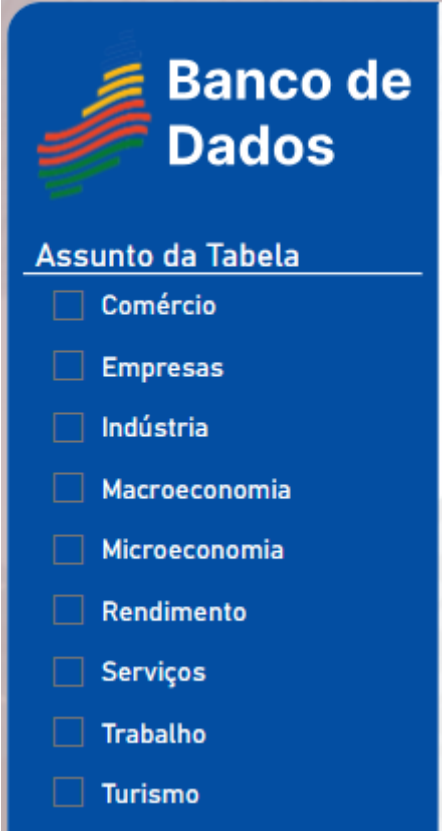


Figure 1: Catálogo de dados - Estudo de emprego e renda no Piauí

Em “assunto da tabela” pode-se filtrar o banco de dados pelos seguintes assuntos:

- Comércio
- Empresa

- Indústria
- Macroeconomia
- Microeconomia
- Rendimento
- Serviços
- Trabalho
- Turismo



Banco de Dados

Assunto da Tabela

- ☐ Comércio
- ☐ Empresas
- ☐ Indústria
- ☐ Macroeconomia
- ☐ Microeconomia
- ☐ Rendimento
- ☐ Serviços
- ☐ Trabalho
- ☐ Turismo

Figure 2: Filtro 1

O banco de dados trabalha com pesquisas que tiveram o período de divulgação entre 2016 e 2024. As tabelas também podem ser filtradas por período de divulgação, ou por tipo de período (mensal, trimestral e anual).

Período de Divulgação

✓

☐

2016

✓

☐

2017

✓

☐

2018

✓

☐

2019

✓

☐

2020

✓

☐

2021

✓

☐

2022

✓

☐

2023

✓

☐


2024


Tipo de Período


☐Ano

☐Mês

☐Trimestre







No campo pesquisa, é possível digitar o nome da tabela, variável, classificador, assunto, categoria ou ano de atualização desejado.

Pesquisa

Tabela

☒Tabela

☐Variáveis


☐Classificadores


☐Assunto

☐Categorias

☐Ano de Atualiz...

Search





da Tabela

Figure 3: Filtro 3

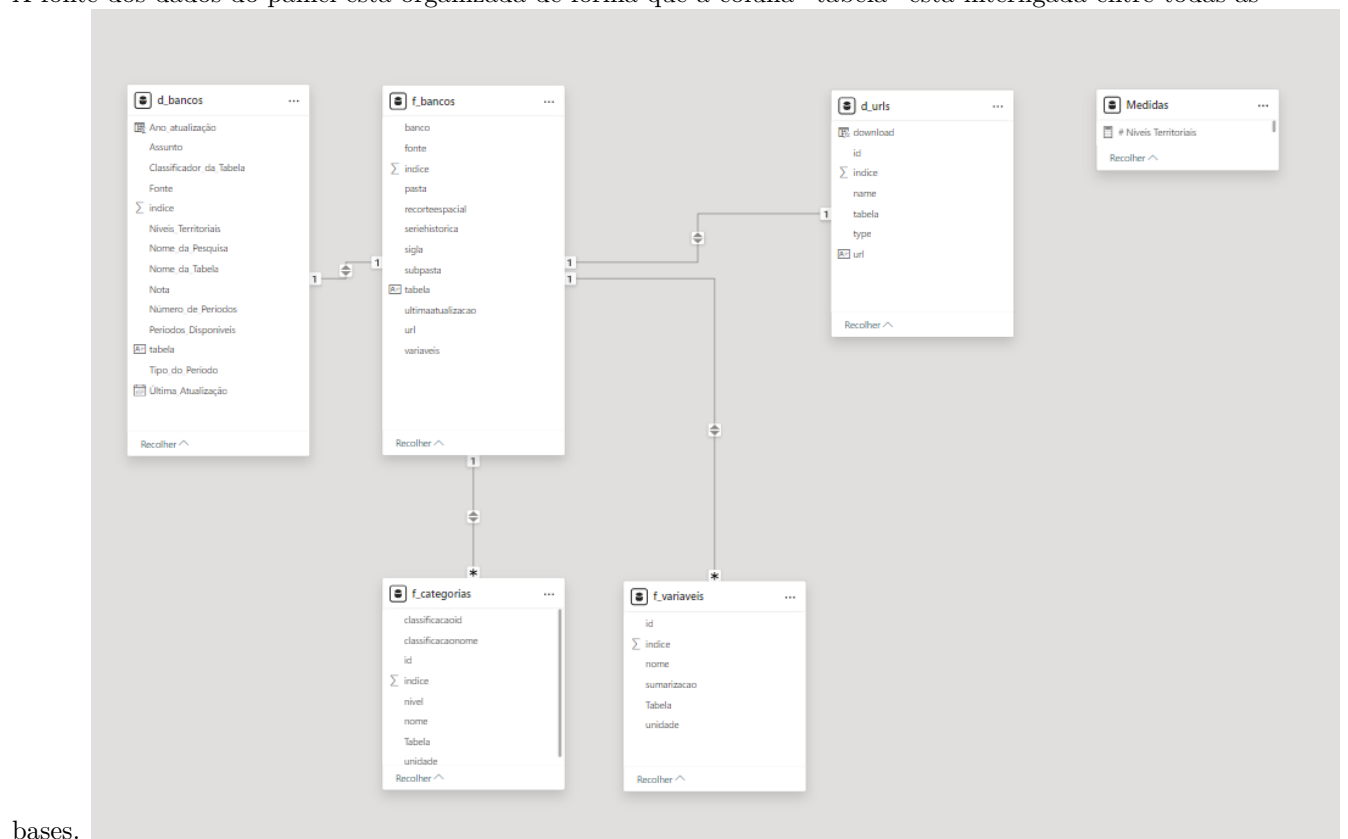
Feitos os filtros desejados, para fazer o download da tabela escolhida basta clicar no símbolo da url.

Tabela	Descrição	Periodicidade	Períodos	Atualização	Download
1209	Índice e variação da receita nominal e do volume de vendas de veículos, motocicletas, partes e peças (2022 = 100)	Ano	13	22/12/2023	🔗
1263	Produção física industrial, por tipo de índice e categorias de uso (série encerrada em fevereiro de 2014)	Mês	48	26/10/2016	🔗
1264	Produção física industrial, por insumos típicos da construção civil e tipo de índice (série encerrada em fevereiro de 2014)	Mês	48	26/10/2016	🔗
1265	Produção física industrial, por tipo de índice e seções e atividades industriais (série encerrada em fevereiro de 2014)	Mês	48	26/10/2016	🔗
1266	Produção Física Industrial, por grupos e classes industriais selecionados (série encerrada em janeiro de 2022)	Mês	48	26/10/2016	🔗
1594	Número de famílias e Tamanho médio da família por grupos de idade da pessoa de referência da família	Ano	1	10/11/2022	🔗

Figure 4: Download de tabela

Modelagem dos dados do painel

A fonte dos dados do painel está organizada de forma que a coluna “tabela” está interligada entre todas as



Autores

Alexandre Barros - [alexand7e] <https://github.com/Alexand7e/>

Gustavo Carvalho - [gustavo-PI] <https://github.com/gustavo-PI>