UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Instituto de Ciência e Tecnologia
Câmpus de Sorocaba

CAIO FERNANDES CHAVES MAXIMIANO

# USO DE TECNOLOGIAS DE *BIG DATA* PARA PROCESSAMENTO E ANÁLISE DE DADOS DA ÁREA DA SAÚDE DO ESTADO DE SÃO PAULO

Sorocaba

2023

CAIO FERNANDES CHAVES MAXIMIANO

# USO DE TECNOLOGIAS DE *BIG DATA* PARA PROCESSAMENTO E ANÁLISE DE DADOS DA ÁREA DA SAÚDE DO ESTADO DE SÃO PAULO

Trabalho de graduação apresentado ao Instituto de Ciência e Tecnologia, da Universidade Estadual Paulista "Júlio de Mesquita Filho" Campus Sorocaba, como parte dos requisitos para obtenção do grau de bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Márcio Alexandre Marques

Sorocaba

2023

Maximiano, Caio Fernandes Chaves

M464u

Uso de tecnologias de big data para processamento e análise de dados da área da saúde do estado de São Paulo / Caio Fernandes Chaves Maximiano. -- Sorocaba, 2023

59 p. : il., tabs., fotos

Trabalho de conclusão de curso (Bacharelado - Engenharia de Controle e Automação) - Universidade Estadual Paulista (Unesp), Instituto de Ciência e Tecnologia, Sorocaba

Orientador: Márcio Alexandre Marques

1. Big Data. 2. Spark. 3. Dashboard. 4. CNES. 5. SQL. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto deCiência e Tecnologia, Sorocaba. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Instituto de Ciência e Tecnologia
Câmpus de Sorocaba

# USO DE TECNOLOGIAS DE BIG DATA PARA PROCESSAMENTO E ANÁLISE DE DADOS DA ÁREA DA SAÚDE DO ESTADO DE SÃO PAULO

CAIO FERNANDES CHAVES MAXIMIANO

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO PARTE DO REQUISITO PARA A OBTENÇÃO DO GRAU DE **BACHAREL EM ENGENHARIA DE CONTROLE E AUTOMAÇÂO**

Prof. Dr. Everson Martins

Coordenador

**BANCA EXAMINADORA*:***

Prof. Dr. MÁRCIO ALEXANDRE MARQUES
Orientador/UNESP-Campus de Sorocaba

Prof. Dr. EVERSON MARTINS
UNESP- Campus de Sorocaba

Dr. THIAGO GONÇALVES DOS SANTOS MARTINS
UNIFESP

Maio de 2023

## AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus por ter me guiado até o presente momento.

Agradeço aos meus pais Lilian e Marcelo por sempre terem me incentivado em vários momentos e decisões da minha vida.

Agradeço também a todos os amigos e professores que fizeram parte do meu caminho, sendo determinantes no meu crescimento pessoal e profissional. Por fim, agradeço ao Prof. Dr. Márcio Alexandre Marques pela orientação e apoio no desenvolvimento deste trabalho.

## RESUMO

CAIO, F. C. M. **Uso de tecnologia de *Big Data* para processamento e análise de dados da área da saúde do estado de São Paulo**. Trabalho de Graduação – Instituto de Ciência e Tecnologia do Campus de Sorocaba, Universidade Estadual Paulista "Júlio de Mesquita Filho", Sorocaba, 2022.

Atualmente, indicadores vêm se tornando cada vez mais utilizados como forma de avaliação de um determinado serviço ou processo, e através deles é possível realizar análises até então desconhecidas, possibilitando a melhoria na gestão dos mesmos. Na área da saúde, o uso de tais indicadores se faz necessário para o auxílio na tomada de decisão de acordo com algum cenário que demande atuação do setor público ou privado. Diante disso, este trabalho visa o desenvolvimento de um processo (*pipeline)* responsável por extrair e analisar os dados do Cadastro Nacional de Estabelecimentos de Saúde (CNES) para o estado de São Paulo. O processo aqui desenvolvido conta com a utilização de ferramentas de *Big Data* para o processamento dos dados, como *Spark* para realização de tratamento e agregações dos dados, a aplicação *online Jupyter Notebook* que permite a criação e edição de *scripts* em diferentes linguagens, além do banco relacional PostgreSQL, local final de armazenamento dos dados tratados. Para a interface com o usuário final foi criado um *dashboard* interativo com a ferramenta *Power BI*, com gráficos e indicadores numéricos da distribuição de serviços e profissionais em todo o estado de São Paulo. Após todas as etapas desenvolvidas foi gerado um processo responsável por atualizar mensalmente uma base de dados utilizada como fonte para um relatório do *Power BI*. Neste relatório é possível verificar o cenário atual do sistema de saúde e serviços em todo o estado de São Paulo, bem como identificar quais regiões necessitam de alguma atuação por parte do poder público, como por exemplo, incentivar a contratação de médicos ou agentes comunitários de saúde.



**Palavras-chave**: *Big Data*; Spark; *dashboard;* CNES; SQL.

## ABSTRACT

CAIO, F. C. M**. Use of Big Data technology for data processing and analysis in the health area of the state of São Paulo**. Undergraduate Work – Institute of Science and Technology, São Paulo State University "Júlio de Mesquita Filho", Sorocaba, 2022.

Nowadays, indicators are becoming more and more used as a way of evaluating a given service or process, and through them it is possible to perform analyzes hitherto unknown, enabling improvement in their management. In the health area, the use of such indicators is necessary to aid in decision-making according to a scenario that requires action by the public or private sector. Therefore, this work aims to develop a process (pipeline) responsible for extracting and analyzing data from the National Register of Health Establishments (CNES) for the state of São Paulo. The process developed here relies on the use of Big Data tools for data processing, such as Spark for data processing and aggregation, the Jupyter Notebook online application that allows the creation and editing of scripts in different languages, in addition to the database relational PostgreSQL, final storage location for processed data. For the end user interface, an interactive dashboard was created using the Power BI tool, with graphs and numerical indicators of the distribution of services and professionals throughout the state of São Paulo. After all the developed steps, a process responsible for monthly updating a database used as a source for a Power BI report was generated. In this report, it is possible to verify the current scenario of the health system and services throughout the state of São Paulo, as well as identify which regions need some action by the public authorities, such as, for example, encouraging the hiring of doctors or community health agents. health.



**Keywords**: Data; spark; health; processing; Sql

## LISTA DE FIGURAS

## LISTA DE TABELAS

**SUMÁRIO**

# 1 INTRODUÇÃO

Em um país de dimensões continentais como o Brasil, a gestão de serviços prestados à população se torna um grande desafio. Ainda que a destinação de recursos não seja levada em consideração, a existência de regiões muito afastadas de grandes centros urbanos provoca a falta de interesse de mão de obra qualificada para prestação de serviços básicos na área da saúde. Esse descompasso entre a demanda e a oferta de médicos no Brasil tem relação com a falta de recursos e de estrutura para trabalhar nas cidades menores, valores de salários e outras garantias (GUIMARÃES, 2018).

Entre 2015 e 2020, a taxa de crescimento de médicos no Brasil foi de 25,1%, enquanto a taxa de crescimento da população foi de 4,8%. Tal crescimento acelerado da população de médicos ocorreu em períodos subsequentes à maior abertura de cursos e vagas de graduação em medicina (SCHEFFER, 2023). Enquanto há transformações em aspectos demográficos da profissão de médico, permanece a desigualdade de distribuição de profissionais no território brasileiro, mesmo após os primeiros reflexos do salto quantitativo de médicos devido à abertura de novas vagas de graduação médica na última década (SCHEFFER, 2018).

Ao olhar para o estado de São Paulo, ainda que se tratando do estado com mais recursos do país, a falta de uma distribuição homogênea de profissionais e serviços de saúde pode ser notada quando analisadas as diferentes regiões, onde os dados apontam que a capital de São Paulo tem 2,4 vezes mais médicos por habitante do que no interior do estado. Tal quadro é responsável por afetar principalmente a população em situação de vulnerabilidade social, que depende de meios públicos de transporte para sua locomoção até o especialista mais próximo, ou até mesmo a falta de serviços básicos como aplicação de vacinas e exames de rotina (GROSSI, 2018).

Diante do exposto, esse trabalho visa, através da utilização de tecnologias de processamento de grandes volumes de dados (*Big Data*), a geração de um estudo analítico a partir das informações disponibilizados na plataforma do Cadastro Nacional de Estabelecimentos de Saúde para o estado de São Paulo, com o objetivo de analisar a distribuição de profissionais e serviços de saúde por todo o estado paulista.

Será implementada uma interface gráfica e interativa, onde o usuário final terá a possibilidade de selecionar um cenário específico que deseja analisar a partir do conjunto de

dados. Dentre as possibilidades de seleção está a localização, especialidade, se atende o SUS ou não, entre outras.

A interface final da ferramenta pode ser utilizada para averiguar a falta de determinados serviços e/ou profissionais em uma certa região, evidenciando um gargalo da gestão pública da saúde em certos casos, e possibilitando assim, um panorama das áreas a serem atendidas pelo poder público.

## 1.1 OBJETIVO

O objetivo deste Projeto Final de Curso (PFC) é, por meio de tecnologias de processamento para grandes volumes de dados (*Big Data)*, desenvolver um *pipeline* de dados responsável por realizar a coleta e o tratamento do conjunto de informações disponibilizado pelo CNES, possibilitando assim, por meio de *dashboards*, a análise de dados da área da saúde para todo o estado de São Paulo, evidenciando áreas deficitárias de algum tipo de serviço ou profissional da saúde, além de mostrar um panorama geral da distribuição de estabelecimentos pelo estado.

# 2 REVISÃO BIBLIOGRÁFICA

## 2.1 BIG DATA

O termo *Big Data* consiste em um grande volume de dados, que apresentam ampla variedade de fontes, podendo ser tanto estruturados a exemplo dos bancos de dados relacionais, como não estruturados, a exemplo de dados em formato de vídeo, texto, áudio e redes sociais. Para Taurion (2013), *big data* é um conjunto de tecnologias, processos e práticas que permitem às empresas analisarem dados a que antes não tinham acesso e tomar decisões ou mesmo gerenciar atividades de forma muito mais eficiente (TAURION, 2013).

Ainda de acordo com Taurion (2016), o termo *Big data* pode ser exemplificado em nosso cotidiano através dos 5 V´s:

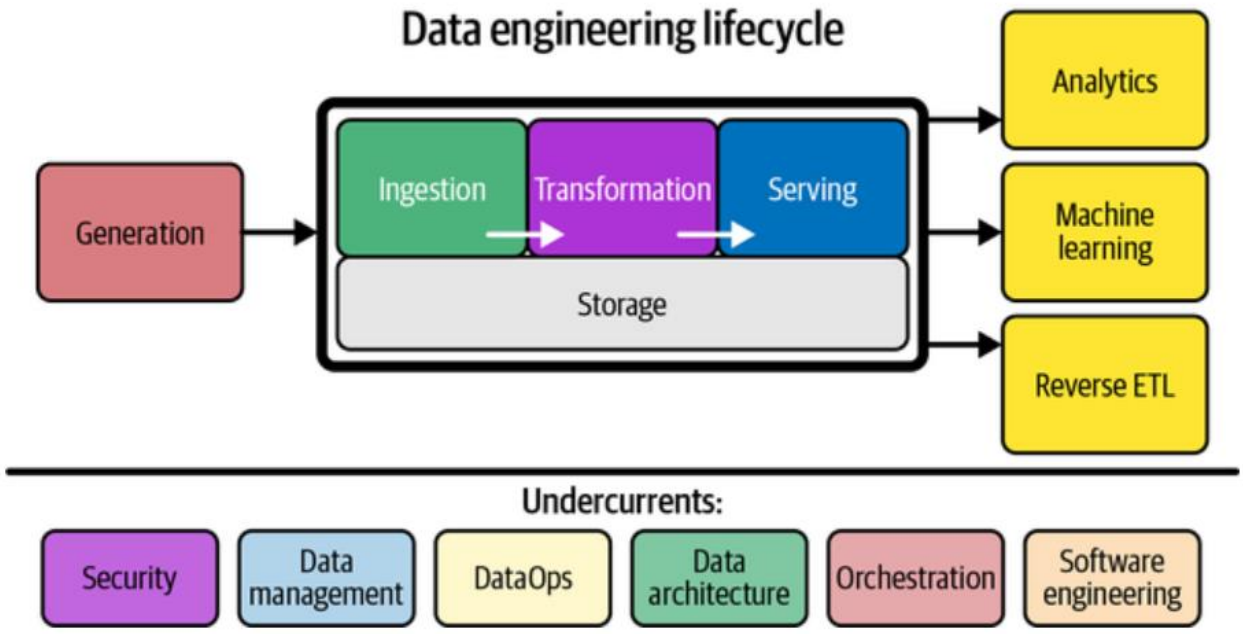
- Volume: grande variedade de dados gerada diariamente.
- Variedade: dados provenientes de inúmeras fontes, com os mais variados formatos.
- Velocidade: existe a necessidade de se processar tais volumes de dados de maneira rápida, algumas vezes em tempo real.
- Veracidade: é de extrema importância que o dado seja confiável para sua utilização nos processos diários.
- Valor: é necessário que se possa obter valores através dos dados, para que as empresas os utilizem para auxílio nas tomadas de decisões (TAURION, 2016).

## 2.2 ENGENHARIA DE DADOS

Um dos termos que mais têm sido comentado nas empresas de tecnologia ao redor do mundo é a engenharia de dados, segmento de trabalho que têm se tornando muito evidente nos últimos anos por trabalhar diretamente com a grande quantidade de dados gerada diariamente pelos diferentes sistemas. De acordo com Reis (2022), a engenharia de dados é o desenvolvimento, implementação e manutenção de sistemas e processos que recebem dados brutos e produzem informações consistentes e de alta qualidade que dão suporte a casos de uso *downstream*, como análise e aprendizado de máquina.

Ainda de acordo com Reis (2022), a engenharia de dados é a interseção de segurança, gerenciamento de dados, arquitetura de dados, orquestração e engenharia de software. Um engenheiro de dados gerencia o ciclo de vida da engenharia de dados, como exemplificado na figura 1 começando com a obtenção de dados dos sistemas de origem e terminando com o fornecimento de dados para casos de uso, como análise ou aprendizado de máquina (REIS, 2022).

Figura 1 - Ciclo de engenharia de dados.




Fonte: Reis, 2022

### 2.2.1 Etl

Um dos processos que mais tem se tornado comum nas empresas devido ao grande volume de dados provenientes de diferentes fontes é o ETL (*Extract, Transform and Load)*. De acordo com Mahmmod (2021), a processo de ETL consiste na movimentação dos dados em três diferentes níveis, de uma ou mais fontes em direção a um destino.

Com relação a **extração**, podemos imaginar sistemas legados (sistemas ou softwares que já estão ultrapassados) de empresas, ou seja, plataformas e softwares obsoletos que estão em uma companhia por muitos anos, geralmente responsáveis por absorver processos transacionais do dia

a dia da empresa, mas que por sua vez não fornecem visões analíticas, sendo necessário como primeiro passo a extração de tais conjuntos de informações aplicando as regras de negócio.

Uma vez que as informações são retiradas de diferentes fontes de uma companhia, eles ainda não estão preparados para serem analisados, precisam de um conjunto de transformações e agregações para que obtenham uma estrutura passível de utilização.

Após a realização de transformações no conjunto de dados extraídos, o último passo se torna o carregamento de tais informações em uma plataforma onde os analistas e cientistas de dados possam realizar análises, podendo ser, por exemplo, um banco relacional, representando o conjunto de dados em forma de tabelas (MAHMOOD, 2021).

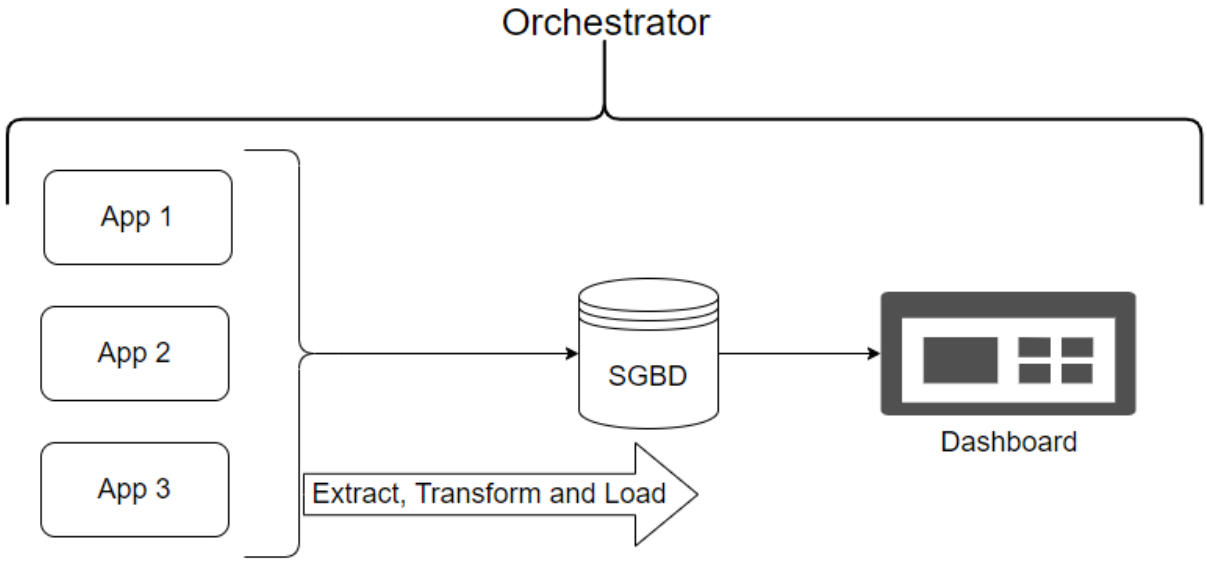
### 2.2.2 Arquitetura de dados

Devido ao grande crescimento do volume de dados nos últimos anos, muitas empresas tiveram que passar por uma transformação com relação a forma que trabalhavam com suas informações, demandando tempo e investimento para tal.

Como forma de suprir essa necessidade, começaram a surgir inúmeras ferramentas de *Big Data* que tinham como objetivo não somente o armazenamento de um grande volume de informações, mas também o processamento dos dados em larga escala e com um tempo muito baixo quando comparado a tecnologias obsoletas.

Diante de tal cenário, surgiu a necessidade de um conjunto de modelos e regras que governem e controlem os dados, conforme exemplificado na figura 2. Ela mostra soluções de integração entre diferentes aplicações, por exemplo o App 1 sendo uma aplicação *web* responsável por enviar pedidos de compra, o App 2 um controle de estoque etc. Todas essas informações seriam armazenadas em um sistema gerenciador de banco de dados (SGBD) para posterior consumo por parte de algum relatório analítico.

Figura 2 - Exemplo de arquitetura de dados para realização de ETL.




Fonte: Autoria Própria.

## 2.3 APACHE SPARK

Como definição geral, Apache Spark pode ser definido como um *framework* de computação de solução rápida e de uso geral para processamento de dados em larga escala. Foi desenvolvido pela AMPLab da Universidade da Califórnia e posteriormente repassado para a Apache Software Foundation, que o mantém desde então.

Se trata de uma ferramenta *open source* e de processamento distribuído para processos na área de *big data* que através de um suporte de processamento em memória aumenta o desempenho de *queries* para grandes volumes de dados quando comparado com processamento em disco.

Com relação ao uso geral de tal solução, temos a possibilidade de rodar múltiplas tarefas em SQL, criação de *pipeline* de dados, inserção de dados em bancos de dados e até mesmo a execução de algoritmos de *Machine Learning*.

Entre as inúmeras vantagens que ajudaram na popularização do Apache Spark nos últimos anos, temos:

- **Integração**: agrega recursos complexos e relevantes como algoritmos de gráfico e aprendizado por reforço, tornando uma solução robusta para várias aplicações que utilizam dados.
- **Velocidade**: uma das características predominantes do Spark foi suprir os 5V´s do *Big Data* através do seu processamento em memória, fornecendo soluções rápidas quando comparadas com outras soluções de dados.
- **Flexibilidade**: Apache Spark suporta múltiplas linguagens e possibilita aos desenvolvedores criar aplicações em Java, Scala, R ou Python.
- ***Streaming* de dados**: o Spark Streaming facilita a criação de fluxos de processamento tolerantes a falhas e com dados em tempo real.
- **Ferramenta *Open Source***: com tal cenário o sistema é apoiado por comunidades globais, de tal forma que novas ideias surgem a todo instante, aperfeiçoando cada vez mais a plataforma.

Segundo Apache (2016), o projeto Apache Spark inclui módulos para aprendizado de máquina (MLlib), processamento em grafos (GraphX), processamento em *stream* (Spark Streaming) e SQL (Spark SQL), sendo possível combinar essas bibliotecas em uma mesma aplicação (SINHA, 2018).

## 2.4 ANACONDA IDE

O Anaconda IDE consiste em uma plataforma de distribuição *open source* de linguagem de programação Python e voltada para projetos de dados. Ela busca simplificar o gerenciamento de informações e a sua utilização.

Tal solução é amplamente utilizada pela disponibilização de bibliotecas e funções que podem ser aplicadas em diversos projetos. Ela oferece todos os pacotes necessários para o desenvolvimento de soluções na área de dados.

Outro grande diferencial do Anaconda IDE que pode ser ressaltada é a existência de ambientes virtuais, que possibilitam um melhor gerenciamento dos pacotes e requisitos necessários para cada projeto que um desenvolvedor possa estar envolvido, como interpretador Python utilizado e bibliotecas (COUTINHO, 2020).

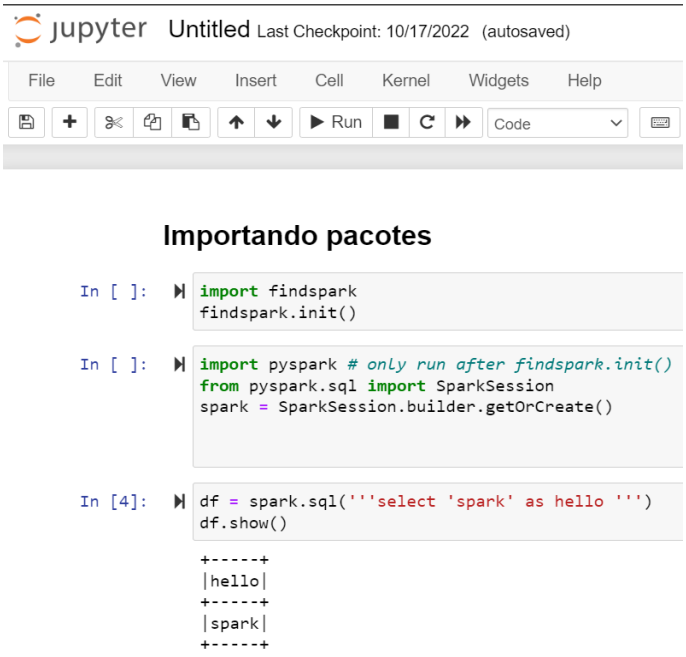
## 2.5 JUPYTER NOTEBOOK

O Jupyter Notebook é um aplicativo gratuito de código aberto que através de uma interface gráfica permite aos seus usuários a edição de *notebooks* em um navegador *web*, como Google Chrome ou Firefox.

Na figura 3 é mostrado o *layout* da aplicação, em diferentes linhas de código executáveis separadamente (em lotes). Através de tal ferramenta é possível realizar o desenvolvimento de códigos responsáveis pela "limpeza", transformação e até visualização de dados, além da criação de arquivos executáveis que podem ser compartilhados entre usuários.

Através de células que podem ser do tipo "marcação" ou "código", o desenvolvedor pode executar os comandos em lotes, não sendo necessária a execução do código de ponta a ponta. Além disso, o Jupyter Notebook fornece suporte para mais de 40 linguagens de programação (HENRIQUE, 2020).

Figura 3 - Estrutura do Jupyter Notebook.




Fonte: Autoria Própria

### 2.5.1 Psycopg2

É uma biblioteca responsável pela conexão entre o banco de dados PostgreSQL e o Python. Tal conexão é de grande funcionalidade, uma vez que é possível realizar operações no banco de dados via *scripts* Python como *update*, *delete*, *insert* (DIAS, 2021).

Na figura 4, está exemplificado, por meio de código, o uso de dessa biblioteca para realização da conexão com um banco de dados PostgreSQL local, através da passagem dos parâmetros de conexão.

Figura 4 - Biblioteca Psycopg2.

```
import psycopg2

conn = psycopg2.connect(database="postgres",
                        user='postgres', password='password',
                        host='localhost', port='5432')
conn.autocommit = True
cursor = conn.cursor()
```

Fonte: Autoria Própria.

### 2.5.2 Wget

O Wget é uma biblioteca que facilita a implementação de *downloads* de arquivos com o Python. Pode ser utilizada para diversos fins, como *web crawling*, *download* de *websites* inteiros de forma recursiva, páginas inteiras de imagens etc. (ARAVIND, 2022).

Através da passagem de uma URL (*Uniform Resource Locator* (localizador uniforme de recursos)) específica do servidor remoto, é possível realizar o *download* de um arquivo *zip* da plataforma do CNES. A figura 5 mostra esse processo.

Figura 5 - Biblioteca Wget.

```
import wget

ftp_path = f"ftp://ftp.datasus.gov.br/cnes/BASE_DE_DADOS_CNES_202208.ZIP"

local_path = "C:/Users/013503631/Documents/"

file = wget.download(ftp_path, local_path)
```

Fonte: Autoria Própria

### 2.5.3 Zipfile

O Zipfile do Python consiste em uma biblioteca responsável por manipular arquivos zipados. Através dele é possível realizar a leitura, escrita, compactação e descompactação de vários arquivos em um diretório local, conforme demonstrado na figura 6 (POZO, 2022).

Figura 6 - Biblioteca Zipfile.

```
import zipfile

if zipfile.is_zipfile('file.zip'):
    with zipfile.ZipFile('file.zip') as item:
        item.extractall('C:/Users/013503631/Documents/')
```

Fonte: Autoria Própria

### 2.5.4 Sys

A biblioteca Sys do Python compreende inúmeras funções e variáveis utilizadas para manipular diferentes partes do ambiente de desenvolvimento. Através dela podemos realizar interações com o interpretador Python utilizado, alocando um erro retornado pelo sistema e armazená-lo em uma variável ou exibi-lo para o usuário (RAI, 2021).

### 2.5.5 Os

A biblioteca OS do Python fornece uma solução simples de uso das funcionalidades que são dependentes do sistema operacional. Através de comandos dessa biblioteca é possível ler ou escrever em um arquivo, manipular estruturas de diretórios ou criar arquivos temporários através de suas funções (ASHWIN, 2021).

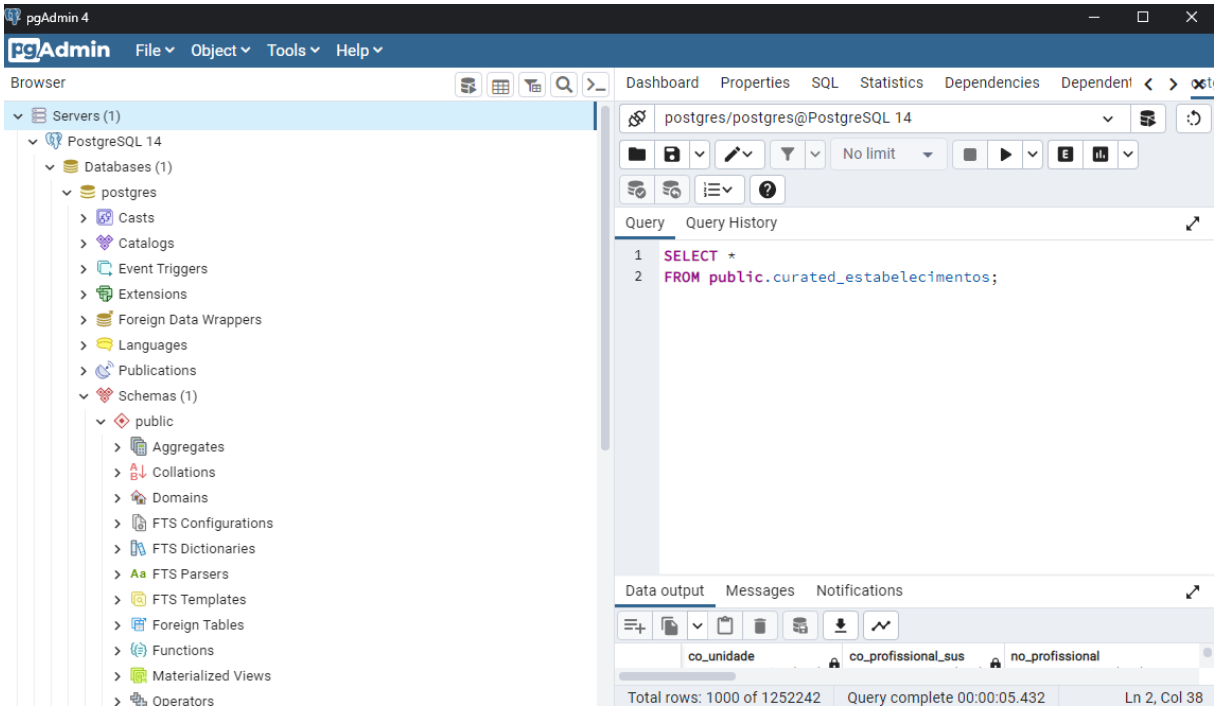
## 2.6 POSTGRESQL

O PostgreSQL consiste em um sistema gerenciador de banco de dados (SGBD) *open-source*, altamente estável e que fornece suporte a diferentes funções de SQL, como chaves estrangeiras, subconsultas, *triggers*, e diferentes tipos e funções definidas pelo usuário. Ele aumenta ainda mais a linguagem SQL oferecendo várias características que meticulosamente escalam e reservam cargas de trabalho de dados. É usado principalmente para armazenar dados para muitos aplicativos móveis, *web*, geoespaciais e analíticas.

Devido a sua fácil integração com ferramentas e diferentes sistemas de empresas, ele se tornou uma das soluções de armazenamento de dados mais populares do mercado. Como principais funcionalidades que tal ferramenta fornece é possível destacar o controle de concorrência multi versionado, recuperação em um ponto do tempo, *tablespaces*, replicação assíncrona, transações agrupadas, um sofisticado planejador de consultas (otimizador) e registrador de transações sequencial (WAL) para tolerância a falhas (BOROZENETS, 2022).

Para o desenvolvimento de consultas e criação de objetos utilizando o PostgreSQL, é possível utilizar o pgAdmin, um software gráfico de administração de SGBD, responsável por oferecer uma interface gráfica amigável para o usuário. A figura 7, mostra o *layout* do pgAdmin, com toda a estrutura de objetos disponíveis para serem trabalhados.

Figura 7 - Layout do administrador pgAdmin.



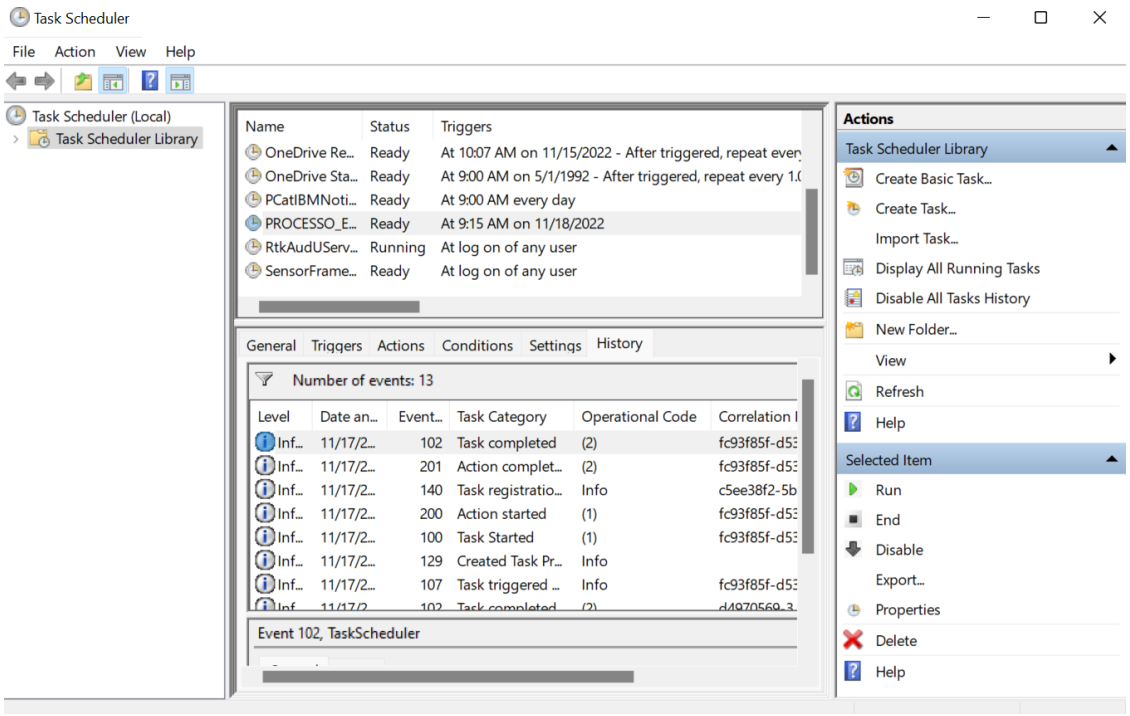

Fonte: Autoria propria.

## 2.7 WINDOWS TASK SCHEDULER

O *Windows Task Scheduler* é uma ferramenta incluída no sistema operacional da *Microsoft* que possibilita a realização de tarefas pré-definidas para que possam ser executadas automaticamente quando algum conjunto de condições é satisfeito. Por exemplo, um dia/horário ou a chegada de algum arquivo em um determinado diretório, são várias as opções responsáveis por disparar o "gatilho" de execução automática da tarefa.

Através de uma interface disponibilizada para o usuário (figura 8), é possível orquestrar a execução de *scripts* diários até então acionados manualmente a cada dia, possibilitando uma automação de processos agendada e parametrizada (THATHA, 2020).

Figura 8 - *Layout* do Windows Task Scheduler.



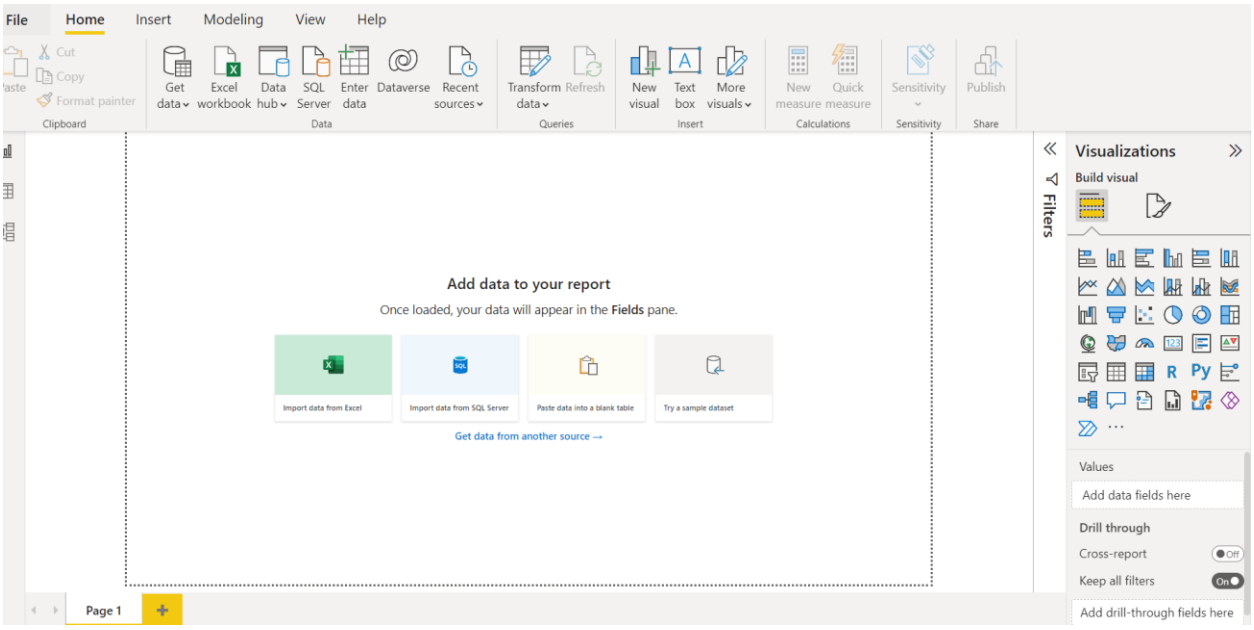

Fonte: Autoria própria.

## 2.8 POWER BI

O Power BI é um software de *Business Intelligence* (BI), ou seja, transforma dados em informação para inteligência de negócios. Essa ferramenta permite ao usuário a conexão com as mais variadas fontes de dados (txt, Excel, csv, banco de dados, *websites* etc.) para extração, gerando indicadores de desempenho e criando *dashboards*.

Com ela é possível realizar a criação de modelos de dados, relacionando diferentes tabelas, tratamento e transformação de dados vindos das mais diferentes origens. Através de uma representação dinâmica das colunas de uma fonte de dados, é possível construir diferentes *dashboards* e indicadores, além de possibilitar a publicação na internet para que diferentes pessoas tenham acesso ao projeto desenvolvido (MAIA, 2021). A figura 9 mostra o a interface Power BI disponível para o usuário e que é muito similar com a de outros produtos fornecidos pela *Microsoft* como o Excel e o Word.

Figura 9 - Layout da página inicial do Power BI




Fonte: Autoria própria.

# 3 MATERIAIS E MÉTODOS

## 3.1 MATERIAIS

Para a execução de todo o *pipeline* de dados foi utilizado um *notebook* HP Elitebook com processador Intel Core i7, memória RAM de 32 GB e sistema operacional Windows 11. Esse dispositivo foi utilizado tanto para a configuração e o desenvolvimento dos *scripts* quanto para o processamento do volume de dados.

Como fonte de dados foi utilizada a base de dados do CNES (Cadastro Nacional de Estabelecimento de Saúde), uma plataforma pública e facilmente acessível com dados da área da saúde atualizados mensalmente.

## 3.2 DICIONÁRIO DE DADOS

Para melhor entendimento dos dados que estavam sendo tratados, bem como a criação de indicadores que melhor representassem o quadro de saúde do estado de São Paulo, foi utilizado o dicionário de dados disponibilizado pelo CNES.

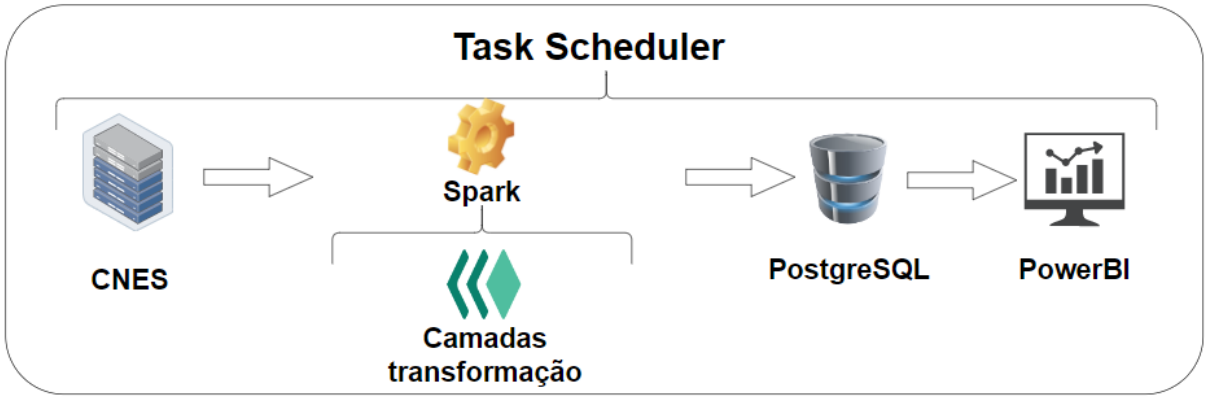
Nesse documento foi possível compreender as chaves que relacionavam cada uma das tabelas disponibilizadas, bem como entender quais agregações deveriam ser feitas para se obter uma tabela final com todos os dados necessários para a criação do *dashboard*. A partir dessa documentação, também foi possível realizar o desenvolvimento de indicadores operacionais, uma vez que estava disponibilizado a descrição dos dados e suas quantidades.

O dicionário de dados com todas as informações, pode ser consultado em ftp://ftp.datasus.gov.br/cnes/Download/SCNES_DICIONARIO_DE_DADOS.ZIP

## 3.3 ARQUITETURA DE DADOS

Para esta aplicação, foi desenvolvida uma arquitetura de dados visando três aspectos principais conforme demonstrados na figura 10: armazenamento, processamento e visualização dos dados. Nessa arquitetura temos o servidor do CNES como principal fonte de dados de saúde para extração inicial. A transformação dos dados é feita pelo *framework* Spark utilizado para "limpeza" e agregação dos dados, juntamente com as pastas locais, que funcionam como camadas. Após as duas primeiras etapas, os dados finais são enviados para o banco de dados PostgreSQL, que serve de armazenamento e fonte de dados para o relatório do Power BI.

Figura 10 - Arquitetura de dados do projeto.




Fonte: Autoria Própria

Inicialmente foi desenvolvida uma arquitetura utilizando serviços em nuvem, mas pela necessidade de pagamento para utilização de recursos necessários, ela foi reformulada para uma solução "*on premise*", trabalhando com armazenamento e processamento na máquina local.

Também foi priorizada a utilização de softwares de código aberto para que seja possível fazer alterações futuras na ferramenta. Para o armazenamento dos arquivos "baixados" da plataforma do CNES, foram criados diretórios locais que simulavam "camadas" dos dados processados.

No aspecto de processamento, foi escolhida a solução através do Apache Spark para processamento do conjunto de dados e geração de dois arquivos finais. Esse conjunto de dados correspondentes a dois arquivos no formato csv, foram inseridos no banco PostgreSQL, possibilitando a visualização e a realização de consultas nos dados transformados.

Através dessa arquitetura foi possível orquestrar a execução de todo processo através do *Windows Task Scheduler*, uma ferramenta da Microsoft disponibilizada aos usuários do sistema operacional responsável por agendar determinadas tarefas. Com isso foi possível automatizar o *download* dos dados da plataforma do CNES mensalmente, sempre após a disponibilização do último conjunto de dados na plataforma.

Ainda através desse orquestrador, foi possível executar o arquivo .py desenvolvido e responsável por criar a sessão *spark* e realizar o processamento dos dados do CNES, realizando as transformações e agregações necessárias para se obter um valor a partir dos dados.

Com tal processamento sendo executado em Spark, também foi possível realizar a integração com o banco de dados PostgreSQL por meio de algumas bibliotecas, permitindo a inserção do arquivo final gerado nas tabelas.

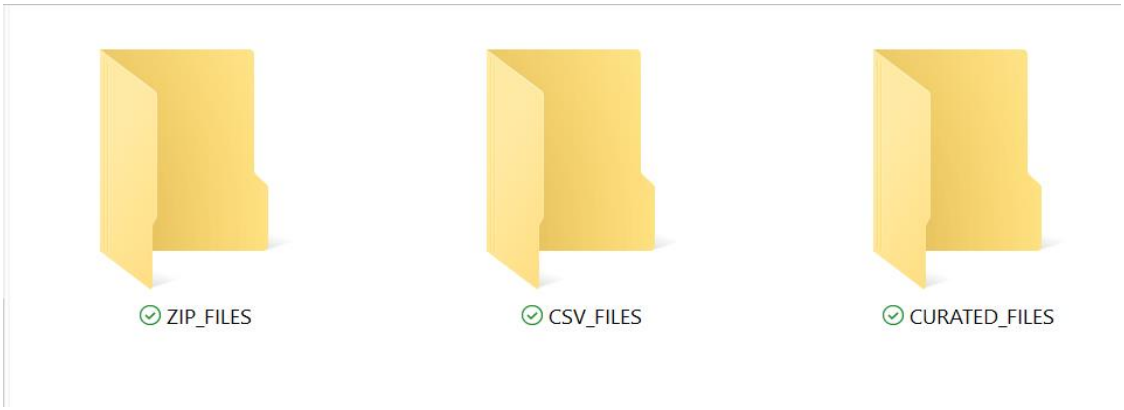
Para visualização do conjunto de dados final pelo usuário, bem como a interação e criação de indicadores, foi utilizado o software Power BI, uma vez que ele fornece integração com o banco de dados PostgreSQL e possibilita a criação de *dashboards* dinâmicos.

## 3.4 CAMADAS DE TRANSFORMAÇÃO

A plataforma do CNES utilizada nesta aplicação como a principal fonte de dados, disponibiliza seu conjunto de informações em arquivos compactados que contêm uma coleção de arquivos csv.

Como forma de organização entre as diferentes etapas de transformação dos dados, foi desenvolvida uma arquitetura de armazenamento em três camadas, como mostra a figura 11. A primeira é a forma bruta com que recebemos os dados (arquivo compactado), a segunda camada é a que contêm os dados já descompactados para serem transformados e a terceira e última, mostra conjunto de informações já agregados e prontos para serem utilizados para geração de indicadores.

Figura 11 - Camadas locais de armazenamento de arquivos.




Fonte: Autoria própria.

No diretório “ZIP_FILES” foram armazenados os arquivos compactados provenientes da plataforma de dados em sua forma bruta. Em um segundo momento, foram extraídos os arquivos com extensão .csv utilizados nas transformações através de uma função, e cujas informações, foram salvas no diretório “CSV_FILES”, que representa a segunda camada da arquitetura, e que serve como repositório para as agregações.

Uma vez que essa coleção de arquivos csv está disponibilizada, foi possível criar a sessão Spark responsável pela leitura, “limpeza” e transformação do conjunto de informações necessárias para geração dos indicadores. Após esse processo, foram gerados dois arquivos finais csv contendo todo o conjunto de informações necessário. Esses dados foram enviados para o diretório “CURATED_FILES”, que representa a terceira e última camada antes de ser inserida no banco PostgreSQL.

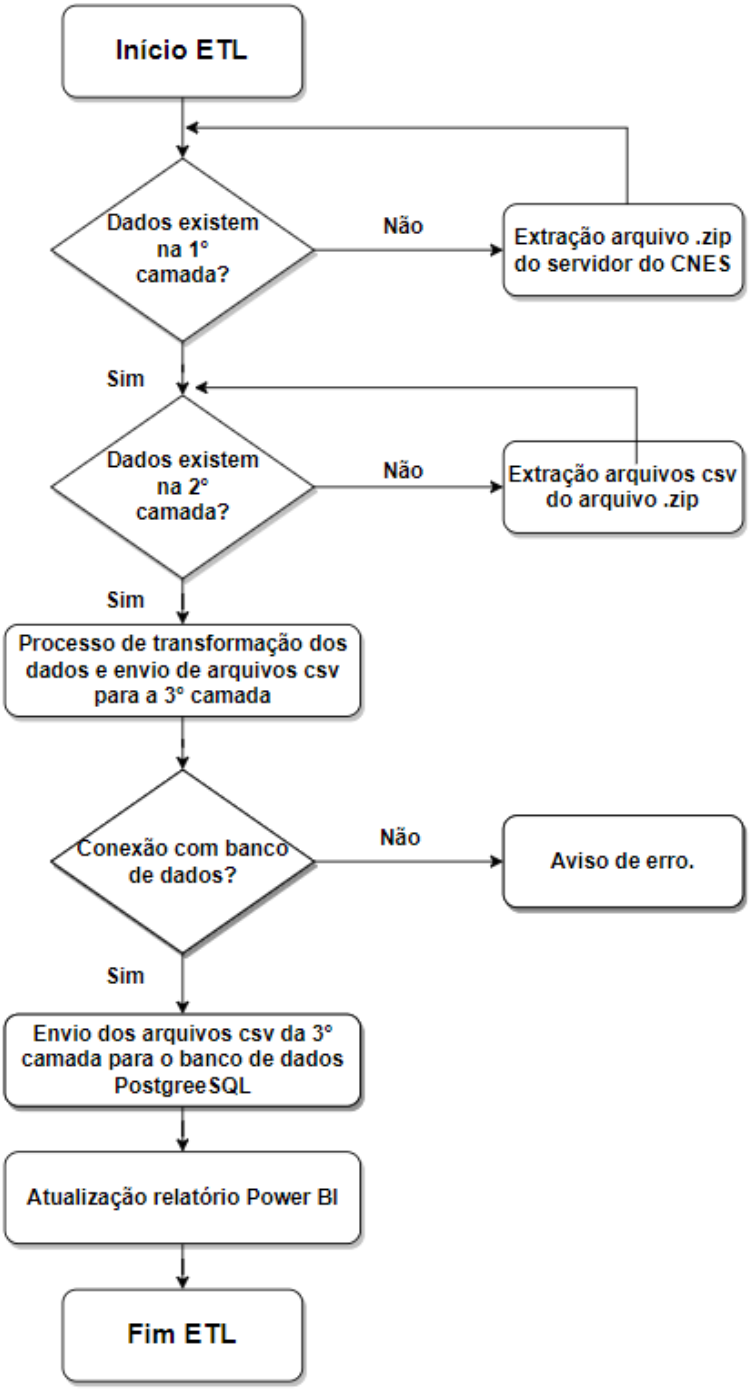
## 3.5 PROCESSO ETL

Para o desenvolvimento do processo de extração, transformação e carregamento, foram utilizados os notebooks (Jupyter Notebbok), possibilitando a criação e execução de determinados “blocos” de código separadamente.

Para o primeiro processo, foram desenvolvidas funções responsáveis pela extração dos dados do servidor do CNES via arquivo compactado e direcionado para a camada “ZIP_FILES”

(figura 12). Uma vez que o arquivo está presente na primeira camada, foi utilizada uma função para extração dos arquivos csv do arquivo compactado, e que, mais tarde, serão utilizados nas transformações.

Figura 12 - Diagrama do processo ETL.




Fonte: Autoria Própria.

Uma vez realizada a extração dos arquivos csv para a segunda camada, toda a etapa de transformação é iniciada. Tal processo consiste basicamente na agregação e "limpeza" dos arquivos disponibilizados pelo CNES, atendendo as regras aplicadas para o processo aqui descrito (como delimitação ao estado de São Paulo).

Após esse processo de transformação ser finalizado, são gerados os dois arquivos finais na terceira e última camada. É possível realizar o envio desses dois arquivos finais gerados para duas tabelas no banco de dados PostgreSQL, assim o relatório do Power BI terá como fonte tais tabelas no banco SQL.

## 3.6 BANCO DE DADOS POSTGRESQL

Uma vez os dados tratados e disponibilizados na terceira e última camada ("CURATED_FILES"), foram criadas as tabelas responsáveis por receber os dois arquivos finais do processo de transformação.

Através do software gráfico para administração do SGBD PostgreSQL foi possível realizar a criação das duas tabelas responsáveis por receber e armazenar os dados de estabelecimentos de saúde e serviços prestados.

A primeira tabela criada foi nomeada como "curated_estabelecimentos", contendo dados dos estabelecimentos de saúde como: o nome dos profissionais que atendem no estabelecimento e sua especialidade, se tal profissional atende ou não o SUS, além informações como nome fantasia, endereço e CEP.

Já a segunda tabela criada foi nomeada como "curated_servicos", contendo os estabelecimentos cadastrados na plataforma e os serviços que cada local disponibiliza para a população. As tabelas 1 e 2 representam a estrutura de ambas as tabelas, bem como as descrições de suas colunas.

Tabela 1 - Estrutura da tabela “curated_estabelecimentos”.

| **Coluna** | **Tipo** | **Descrição** |
|---|---|---|
| co_municipio | integer | Código do município |
| co_cep | integer | Código do CEP |
| data_ingestao | date | Data que foi realizada a ingestão |
| co_cbo | character varying | Código de ocupação segundo CBO |
| tp_sus_nao_sus | character | Atende ou não o SUS |
| ds_atividade_profissional | character varying | Descrição da atividade profissional |
| no_fantasia | character varying | Nome fantasia do estabelecimento |
| co_unidade | character varying | Código do estabelecimento |
| no_municipio | character varying | Nome do município |
| co_sigla_estado | character | Código do estado |
| ds_localidade | character varying | Junção de parâmetros de endereço |
| sk_registro | character varying | Chave auxiliar gerada (*Surrogate Key)* |
| no_bairro | character varying | Nome do bairro |
| co_profissional_sus | character varying | Código do profissional |
| no_profissional | character varying | Nome do profissional |

Fonte: Autoria Própria.

Tabela 2 - Estrutura tabela “curated_servicos”.

| Coluna | Tipo | Descrição |
|---|---|---|
| data_ingestao | date | Data que foi realizada a ingestão |
| co_municipio | integer | Código do município |
| co_servico | integer | Código do serviço |
| co_classificacao | integer | Código da classificação do serviço |
| co_unidade | character varying | Código do estabelecimento |
| no_municipio | character varying | Nome do município |
| ds_classificacao_servico | character varying | Descrição do serviço |
| sk_registro | character varying | Chave auxiliar gerada (*Surrogate Key)* |

Fonte: Autoria própria

Uma vez que as estruturas das tabelas foram criadas fisicamente, a biblioteca “psycopg2” foi utilizada e ela é responsável pela comunicação com o banco PostgreSQL para inserção dos dois arquivos csv localizados no diretório “CURATED_FILES” nas tabelas do PostgreSQL.

Através dessa função, foi possível realizar a inserção dos dados provenientes da camada “CSV_FILES”. Conforme o conjunto de dados for atualizado a cada mês, torna-se necessário a limpeza da tabela e carregamento das informações do mês seguinte.

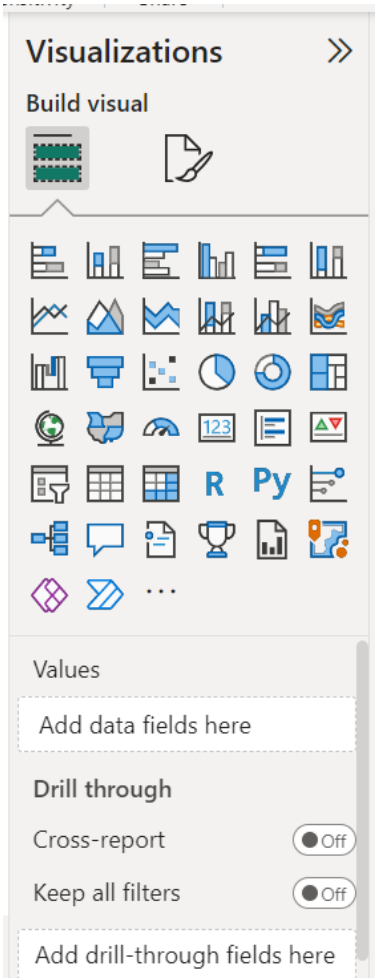
Com as tabelas carregadas com o conjunto de dados tratado e atualizado, foi possível iniciar o desenvolvimento do *dashboard* através do software Power BI.

## 3.7 VISUALIZAÇÃO: INTERFACE E FUNCIONALIDADES

Após todo o processo de ETL finalizado, com os dados carregados nas tabelas “curated_estabelecimentos” e “curated_servicos”, foi iniciado o processo de desenvolvimento da interface com o usuário final através do Power BI.

A figura 13 mostra a disponibilidade de inúmeras visualizações presentes no Power BI, cabendo ao desenvolvedor a escolha de qual tipo se adequada a sua aplicação, fornecendo a melhor visualização dos dados para o usuário.

Figura 13 - Menu de visualizações do Power BI.

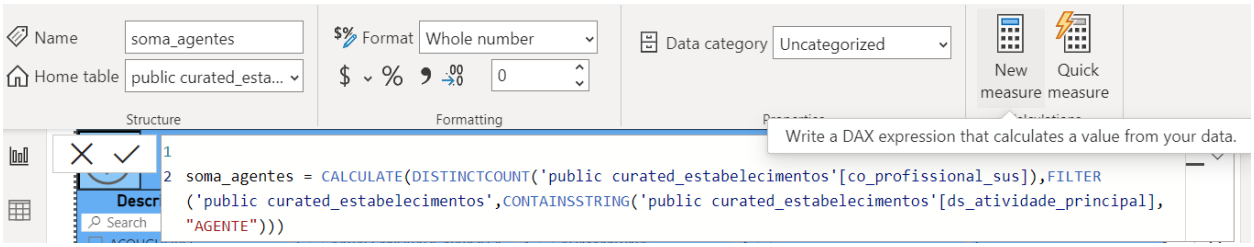



Fonte: Autoria própria.

Através dessa ferramenta foi possível agregar em um único *dashboard* diferentes tipos de visualizações, como gráficos de colunas, gráficos de linhas clusterizado, mapas coropléticos e filtros, tornando a experiência do usuário muito mais intuitiva e possibilitando a geração de valor através de dados brutos.

Também foi possível realizar a criação de indicadores com o Power BI, através da linguagem Dax disponibilizada pela própria ferramenta. Conforme mostrado na figura 14, podemos realizar transformações no conjunto de dados com funções pré-estabelecidas a fim de se atender as demandas de visualização.

Figura 14 - Criação de indicadores com a linguagem DAX.

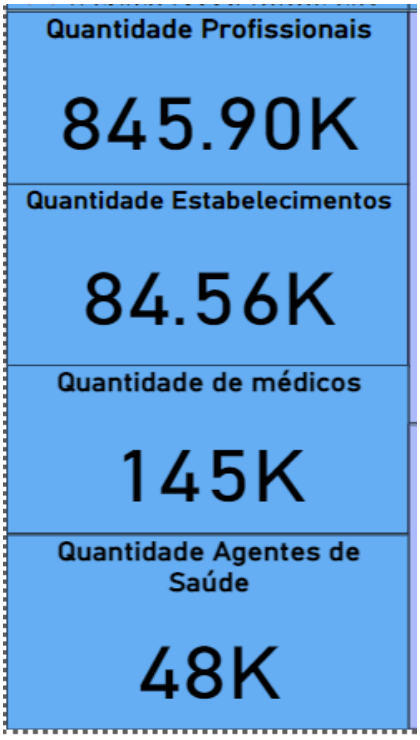



Fonte: Autoria Própria

Uma vez que todo o conjunto de dados foi carregado para o Power BI e que todos os indicadores foram calculados de acordo com as regras definidas, foi iniciado o processo de desenvolvimento do *dashboard*.

Como forma de destacar as informações mais importantes ao usuário, optou-se por utilizar a forma de visualização denominada “cartão”, como pode ser visto na figura 15. Ela consiste em uma única informação, destacada aos olhos do usuário, e geralmente colocada junto de outras informações importantes.

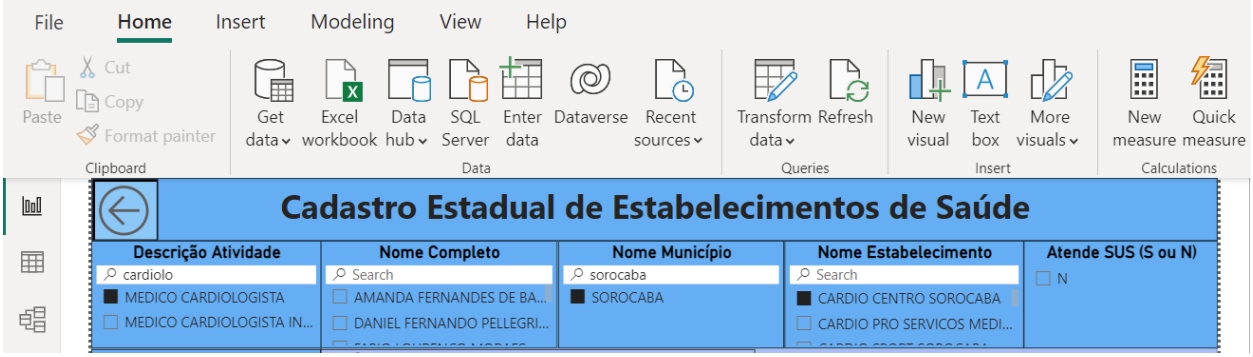
Figura 15 - Cartões de visualização de indicadores.




Fonte: Autoria própria.

Através da interface desenvolvida no Power BI, foi possível a inserção de filtros no *dashboard*. Eles possibilitam a navegação do usuário através de diferentes cenários definidos por ele, tornando assim sua experiência muito mais iterativa e completa do ponto de vista de um *dashboard*. A figura 16 mostra a navegação pelos cenários desejados pelo usuário, bastando selecionar a informação desejada.

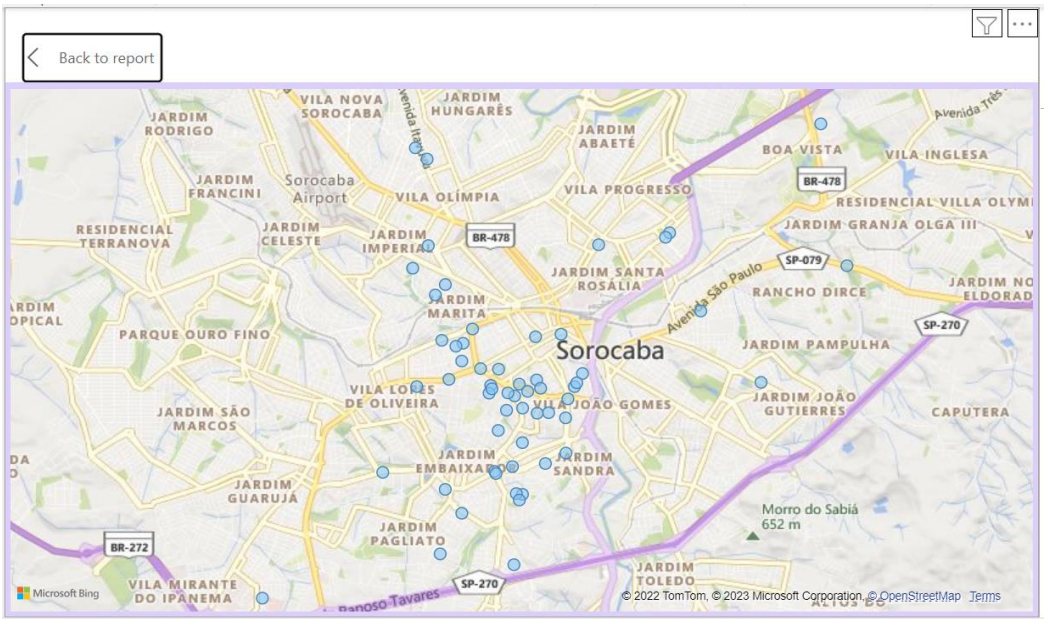
Figura 16 - Filtros de seleção do Power BI.




Fonte: Autoria própria.

Como o conjunto de dados do CNES contem informações de localização para funcionários e serviços, foi desenvolvida uma visualização do tipo mapa, que consiste em uma visão interativa de localização com o usuário. Na figura 17 é possível observar alguns serviços de saúde, conforme selecionados pelo usuário através dos filtros.

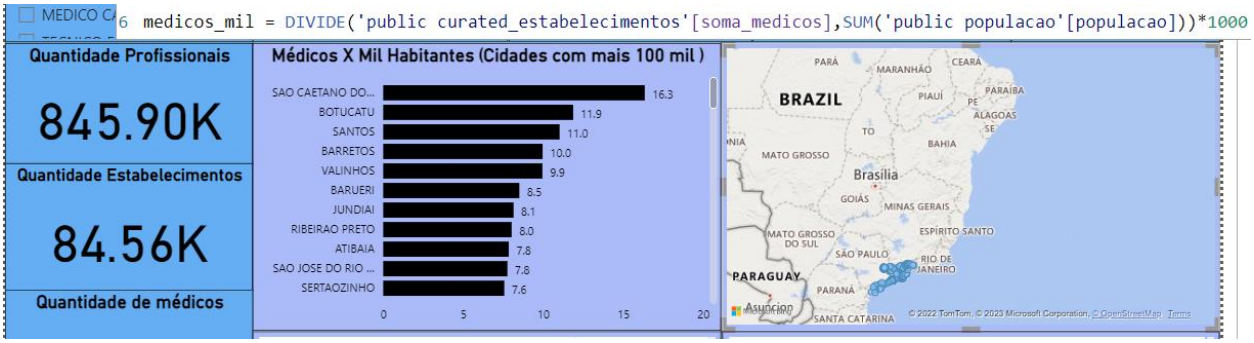
Figura 17 - Mapa do Power BI.




Fonte: Autoria própria.

Também foram criados e agregados à ferramenta alguns indicadores como a quantidade de médicos e agentes comunitários de saúde a cada mil habitantes para cidades com uma população acima de 100 mil pessoas. Com esses indicadores é possível criar visualizações de ranqueamento, oferecendo uma visão ao usuário de quais os melhores locais que atendem tal cenário (figura 18).

Figura 18 - Indicadores de ranqueamento.




Fonte: Autoria própria.

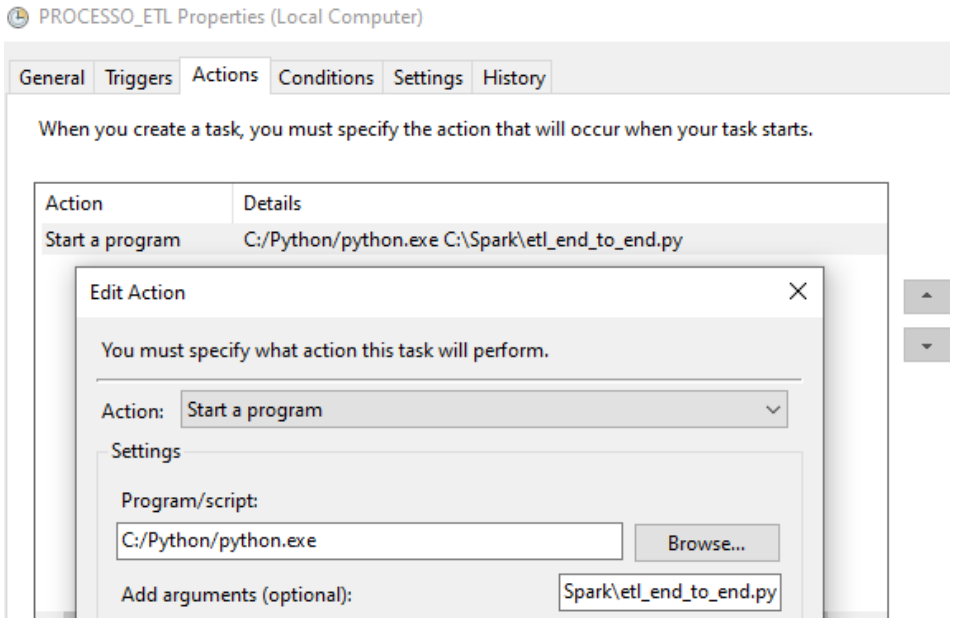
## 3.8 AUTOMAÇÃO DO ETL

Após o desenvolvimento do ETL e a criação do *dashboard* para um certo período, foi necessário realizar a automação do processo para atualização automática do conjunto de dados sem necessidade de uma execução manual.

Para a automação do processo de ETL foi utilizada a ferramenta *Windows Task Scheduler*. Através dela é possível criar uma tarefa que é responsável pela chamada de algum serviço externo, como a execução de um *script* (figura 19).

Apesar da utilização de *notebooks* para o desenvolvimento do processo ETL, também foi necessário implementar um único *script* Python contendo todas as bibliotecas e funções utilizadas, possibilitando o desencadeamento dos processos de extração, transformação e carregamento a partir da chamada de um único *script*.

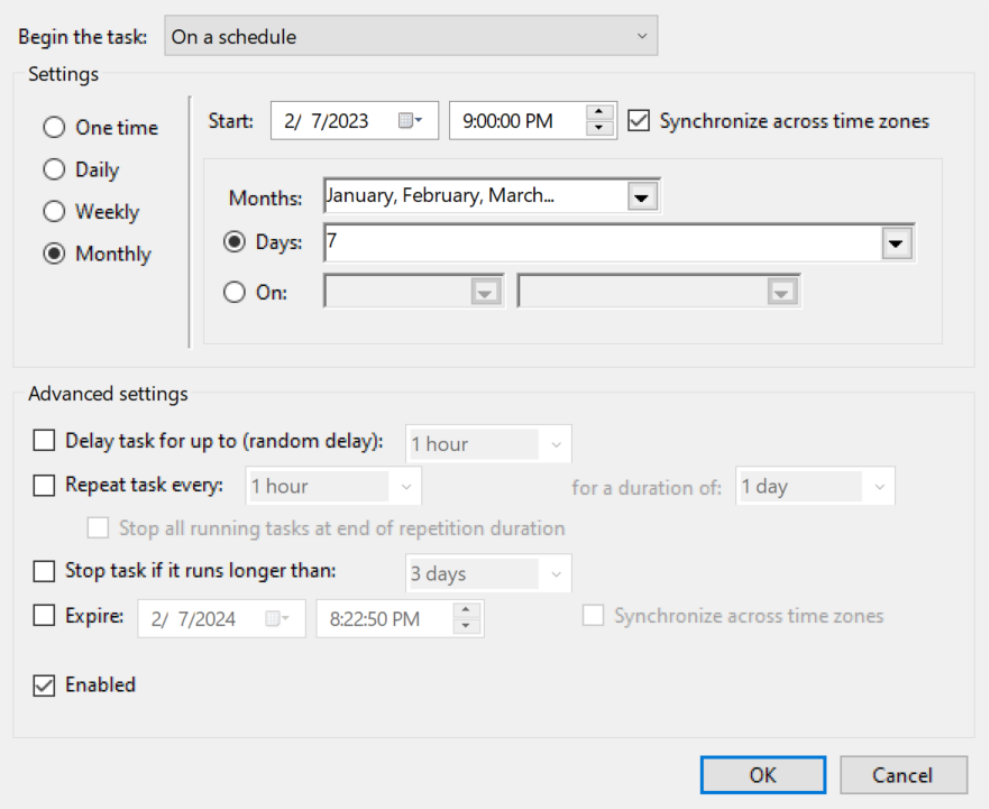
Figura 19 - Implementação da tarefa no Windows Task Scheduler.




Fonte: Autoria própria.

Após o apontamento da tarefa para o *script* responsável pelo processo de ETL, foi utilizada uma interface fornecida pela própria ferramenta para configuração do período de execução (figura 20), ficando estabelecida a execução todo dia 7 de cada mês às 9 horas da noite.

Figura 20 - Painel para definição do dia e hora de execução do script.




Fonte: Autoria própria.

O processo de ETL desenvolvido utiliza dados de dois meses anteriores à data de execução (M-2). Isso se deve ao fato da disponibilização na plataforma CNES. Sendo assim, a tarefa foi configurada para ser executada todo o dia 7 de cada mês, utilizando o conjunto de dados de dois meses anteriores. Os 7 dias após o fim do mês anterior são necessários para a disponibilização dos dados na plataforma do CNES.

Uma vez apontada a tarefa de execução do *script*, bem como a sua configuração para os períodos em que deve ser processada, não se torna mais necessária a ação manual por parte do usuário para que a base de dados seja atualizada pois, todo o processo de ETL fica compreendido no *script* chamado automaticamente pelo *Windows Task Scheduler*.

## 4 RESULTADOS e DISCUSSÕES

Após a etapa de desenvolvimento do processo de ETL e da implementação do *dashboard*, foram trazidos seus resultados para verificação tanto do *pipeline* quanto da interface com o usuário, bem como, seus indicadores formados a partir do conjunto de dados do CNES.

### 4.1 PROCESSO ETL

O processo de ETL realizado após o desenvolvimento consistiu na junção de todos os *notebooks* de extração, transformação e carregamento em um único arquivo Python responsável por inicializar as bibliotecas e funções utilizadas. Assim, foi possível executar a partir de um único *script* Python o processo de ponta a ponta. A figura 21 mostra as mensagens de saída no terminal de execução do *script*, evidenciando o processo até sua parte final de atualização das tabelas no banco PostgreSQL.

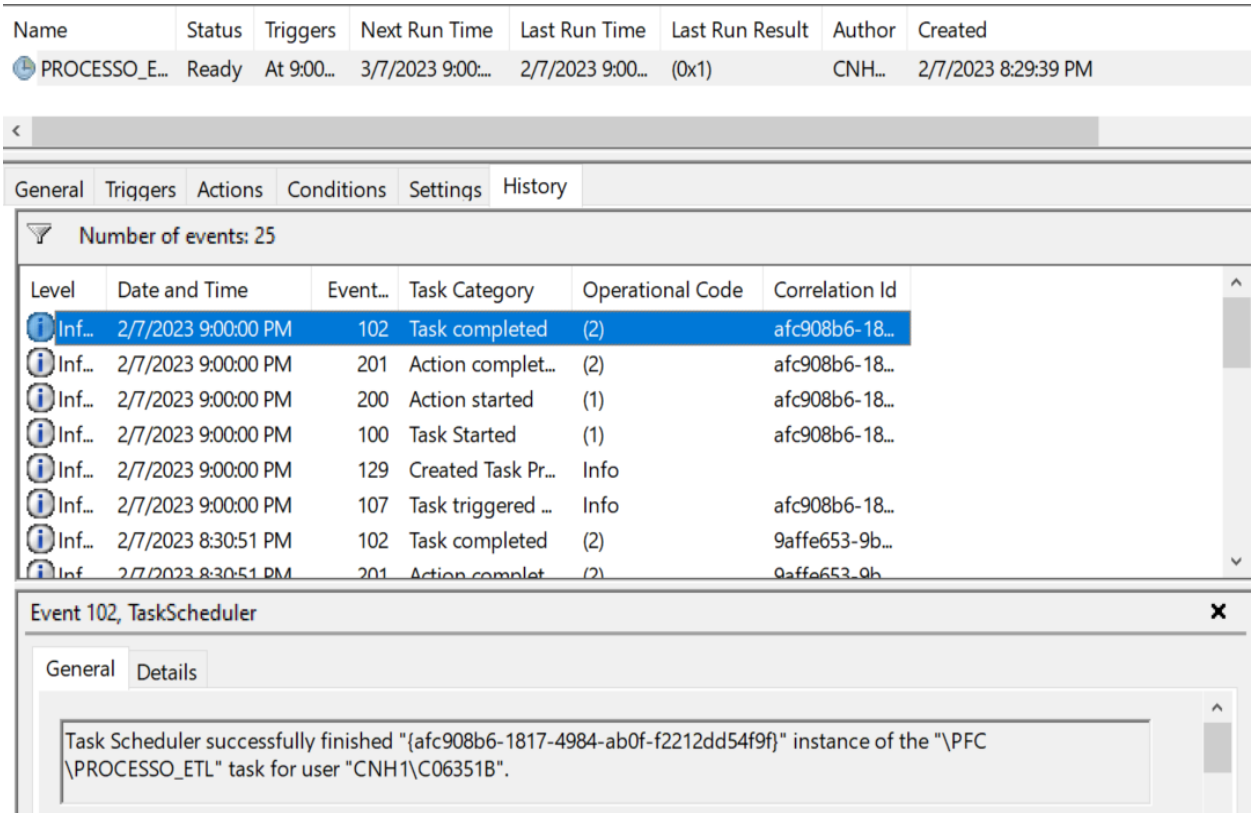
Figura 21 - Execução da carga do processo de ETL.

```
C:\Users\c06351b
> C:/Python/python.exe C:\Spark\etl_end_to_end.py
23/02/07 21:38:40 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform
... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
O arquivos zip de 202211 já existe
Inciando extração de arquivos CSV
A pasta de destino 202211 já existe
23/02/07 21:38:47 WARN package: Truncated the string representation of a plan since it was too
large. This behavior can be adjusted by setting 'spark.sql.debug.maxToStringFields'.
23/02/07 21:38:52 WARN ProcfsMetricsGetter: Exception when trying to compute pagesize, as a res
ult reporting of ProcessTree metrics is stopped
DataFrame curated_estabelecimentos escrito na camada Curated
DataFrame curated_servicos escrito na camada Curated
Tabela curated_servicos atualizada com dados de 202211
Conexão PostgreSQL encerrada
Conexão PostgreSQL encerrada
Tabela curated_estabelecimentos atualizada com dados de 202211
Conexão PostgreSQL encerrada
Conexão PostgreSQL encerrada
```

Fonte: Autoria própria.

Através do Windows Task Scheduler foi possível realizar a chamada do *script* responsável pelo processo ETL, com seu agendamento configurado para o dia 7 de todo o mês. Nenhuma execução manual foi necessária para que a base de dados fosse atualizada, como mostra o histórico de execuções da figura 22.

Figura 22 - Painel de eventos registrados pelo Windows Task Scheduler.



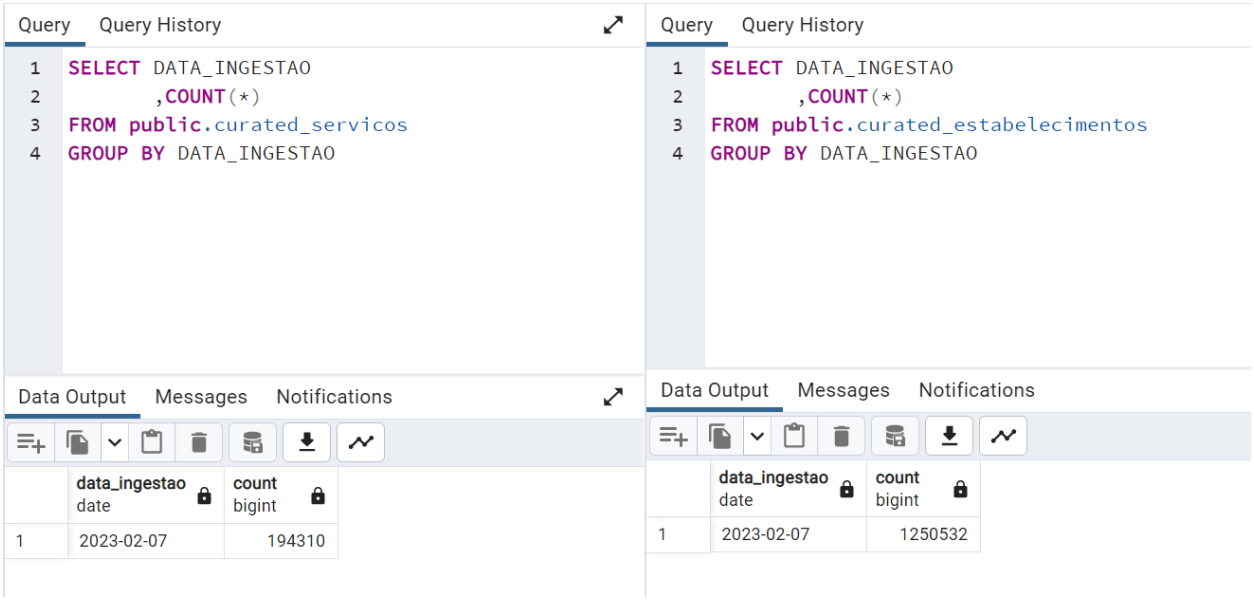

Fonte: Autoria própria.

Como é possível observar, a tarefa criada no Windows Task Scheduler responsável pela execução do *script* Python, foi realizada com sucesso no dia 7, demonstrando a automação do processo sem nenhuma ação por parte do usuário no processo de ETL.

## 4.2 *DASHBOARD*

Uma vez o processo de ETL executando automaticamente via chamada do Windows Task Scheduler, foi possível manter as tabelas do PostgreSQL atualizadas conforme o mês, servindo de fonte de dados para a interface com o usuário. A figura 23 mostra a quantidade de registros em cada uma das duas tabelas, bem como a data de inserção pelo processo de ETL.

Figura 23 - Tabelas do PostgreSQL populadas

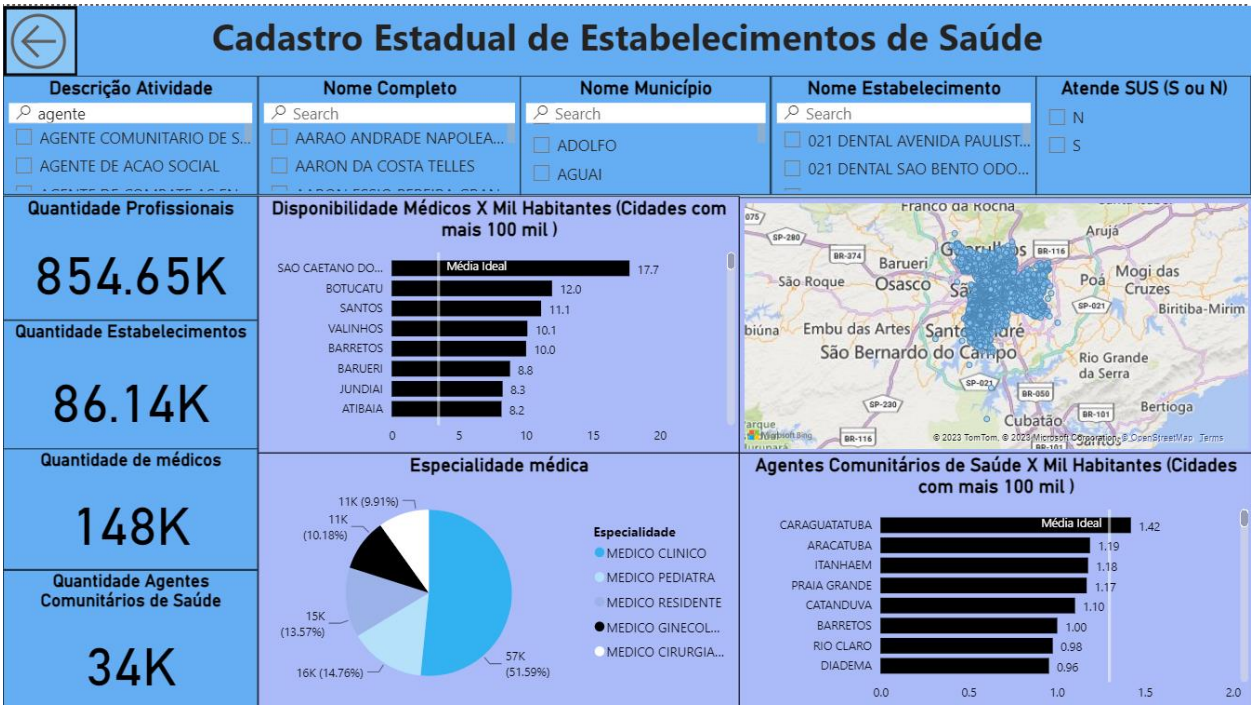



Fonte: Autoria própria.

Após o apontamento do Power BI para as tabelas PostgreSQL com os dados atualizados, foi desenvolvido um *dashboard* para servir de interface com o usuário, como mostra a figura 24, possibilitando o uso de filtros e gráficos de acordo com a aplicação desejada.

Neste ponto, o banco de dados PostgreSQL atua como principal fonte de dados para o *dashboard* utilizado pelo usuário, bastando apenas selecionar o botão “atualizar” para carregar o novo conjunto de dados e recalcular indicadores.

Figura 24 - Painel desenvolvido via Power BI.



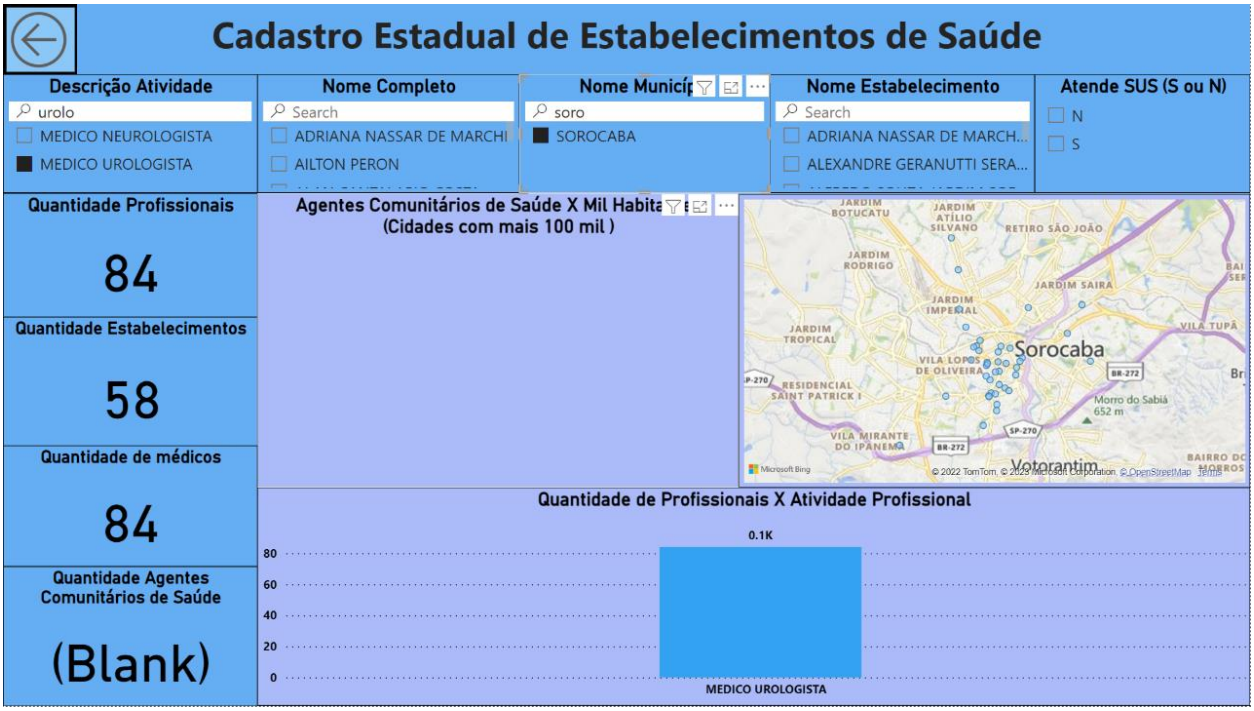

Fonte: Autoria própria.

## 4.3 FILTROS E INFORMAÇÕES CADASTRAIS

Com o auxílio da ferramenta Power BI, foi possível o desenvolvimento de um *dashboard* interativo, onde o usuário pode verificar os dados de estabelecimentos e profissionais da saúde de acordo com a sua região, Especialidade Médica ou até mesmo o profissional desejado.

Como cenário de teste, foi realizada uma busca por médicos Urologistas na cidade de Sorocaba, utilizando para isso, os filtros localizados na parte superior do *dashboard*. Uma vez selecionados de acordo com o cenário desejado, é gerada uma lista com os nomes de profissionais que atendem tal quesito, como mostrado na figura 25.

Figura 25 - Seleção de cenário de testes através dos filtros.

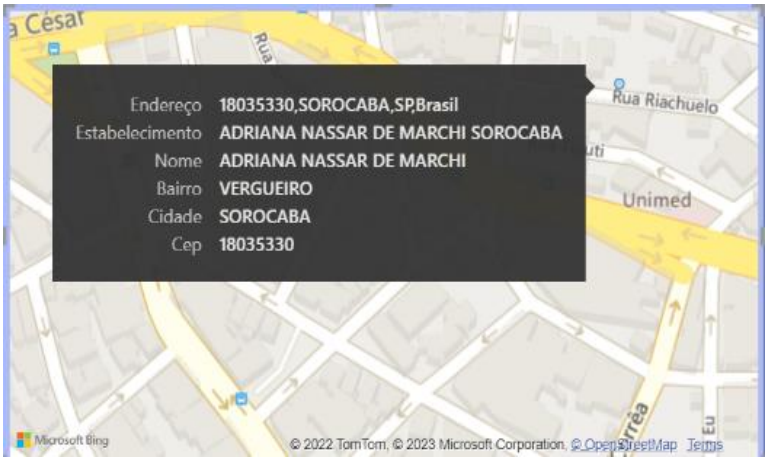



Fonte: Autoria própria.

Uma vez com a lista de 84 médicos Urologistas de Sorocaba distribuídos em 58 estabelecimentos, foi escolhido aleatoriamente a profissional Adriana Nassar de Marchi para verificação dos dados de atendimento do cenário de teste.

Na figura 26 é possível observar as informações dessa profissional selecionada, bem como, seu local de atendimento através do *dashboard*.

Figura 26 - Mapa com informações segundo cenário de teste.




Fonte: Autoria própria.

As informações presentes no mapa do Power BI foram checadas com base no guia médico da página do Conselho Regional de Medicina de São Paulo (GUIA MÉDICO, 2020) para que fosse possível realizar a verificação do cenário de teste (figura 27), simulando algum usuário específico que tivesse a necessidade de achar o local de atendimento para o profissional em questão.

Figura 27 - Informações do médico escolhido segundo cenário de teste.


Fonte: Autoria própria.

Como é possível observar, o local de atendimento da médica escolhida está de acordo com o local mostrado no *dashboard*, tanto o endereço informado quanto o cep indicado, evidenciando a consistência dos resultados das transformações realizadas durante o processo de ETL entre os diferentes arquivos.

## 4.4 INDICADORES E DISTRIBUIÇÃO TOTAL

Uma vez realizado o processo de ETL e disponibilizado o conjunto de dados final, foi possível realizar, através da linguagem DAX, a criação de indicadores qualitativos e quantitativos que possibilitassem ao usuário ter um panorama geral de como os profissionais e estabelecimentos de saúde se distribuem pelo estado de São Paulo tendo como base as tabelas do PostgreSQL.

Assim, para que o usuário tivesse fácil acesso aos números totais dos conjuntos de dados, foram implementadas visualizações do tipo cartão, evidenciando as principais informações numéricas necessárias à primeira vista do usuário ao analisar o *dashboard*. Tais valores numéricos apresentados na figura 28, são automaticamente atualizados conforme o processo de ETL é executado todo dia 7.

Figura 28 - Indicadores quantitativos do Power BI.


Fonte: Autoria própria.

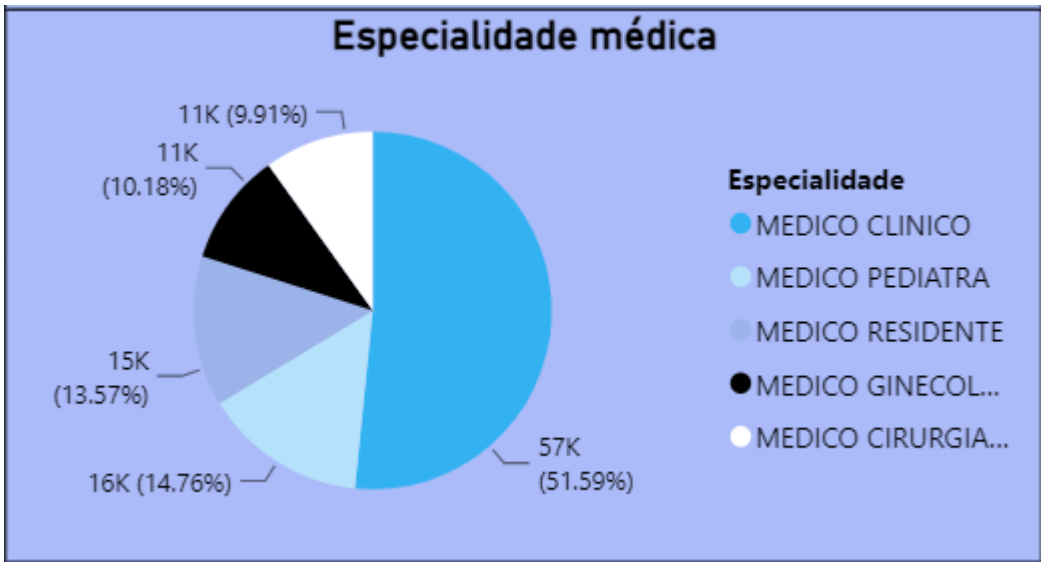
Através desses indicadores, é possível realizar uma análise quantitativa tanto de profissionais e estabelecimentos de saúde, quanto de agentes comunitários de saúde. Esse tipo de visualização também permite obter os indicadores de acordo com o cenário escolhido pelo usuário, como por exemplo, alguma cidade ou Especialidade Médica específica.

### 4.4.2 – Distribuição de profissionais médicos

Por meio dos dados disponibilizados pelo CNES e do processo de ETL desenvolvido, foi possível implementar uma visualização a respeito das principais Especialidades Médicas presentes no Estado de São Paulos, bem como Médicos Generalistas.

Na figura 29 é possível observar a presença majoritária de Médicos Generalistas no estado de São Paulo. Esse fato era esperado, visto que muitas vezes o atendimento primário realizado à população em postos de saúde é realizado por um Clínico Geral.

Figura 29 - Distribuição de médicos no estado de São Paulo.



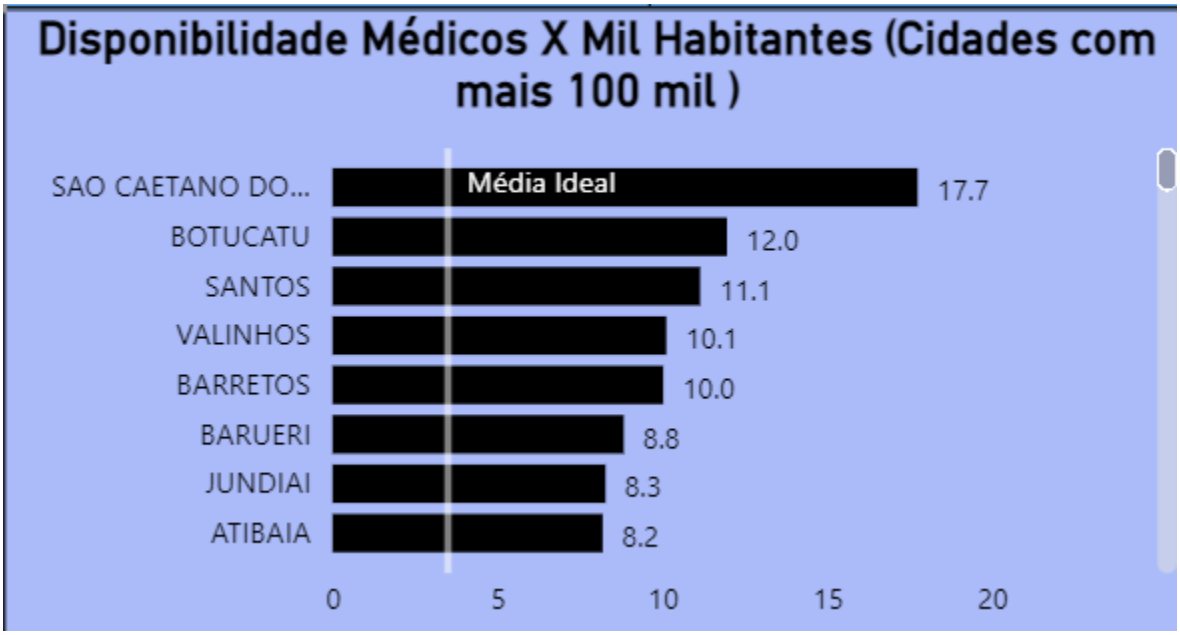

Fonte: Autoria própria.

### 4.4.3 – Médicos a cada mil habitantes

Através do conjunto de dados analisado, foi possível implementar um indicador qualitativo a respeito da disponibilidade de médicos para as cidades com mais de 100 mil habitantes, e que geralmente servem como grandes centros para cidades vizinhas e menores.

Na figura 30 é possível observar um agrupamento de cidades no topo desse *ranking* que também tem um bom índice de desenvolvimento humano ou que possuem universidades de medicina.

Figura 30 - Indicador qualitativo de disponibilidade de médicos.




Fonte: Autoria própria.

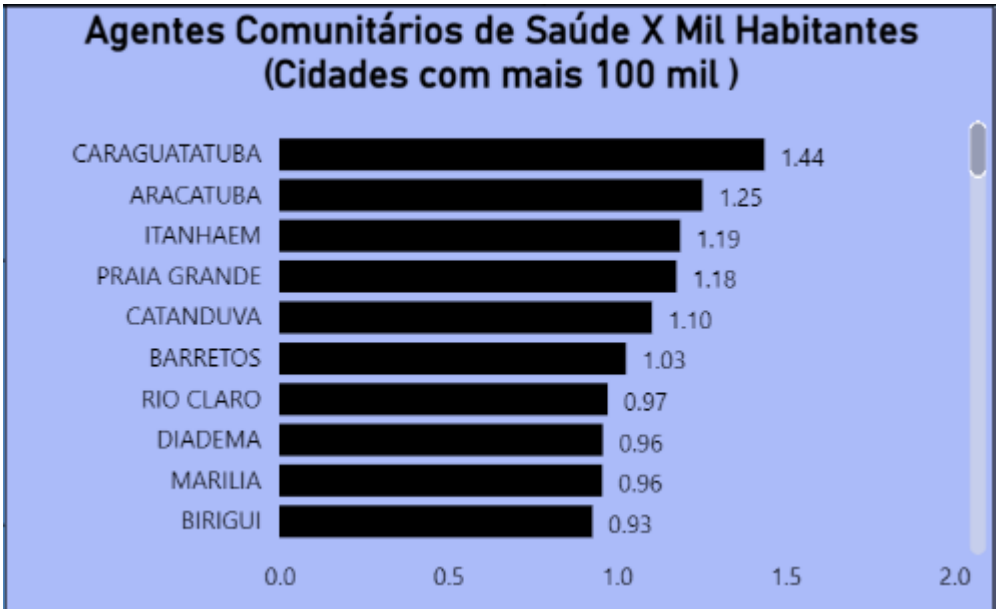
De acordo com a Organização para Cooperação e Desenvolvimento Econômico (OCDE), o número de médicos ideal a cada mil habitantes deve ser de 3,5, valor atingido em muitas cidades do estado de São Paulo, o estado mais rico e que forma mais médicos no Brasil (PORTELLA, 2023).

### 4.4.4 – Agentes Comunitários de saúde a cada mil habitantes

Utilizando a linguagem DAX, disponibilizada pelo Power BI, foi possível implementar um indicador qualitativo a respeito dos Agentes de Saúde Comunitários (ACS) para as cidades com mais de 100 mil habitantes, e que geralmente servem como grandes centros para cidades vizinhas e menores.

De acordo com a Portaria n° 648/GM de 28 de março de 2006 publicado pelo Ministério da Saúde o número indicado de agentes comunitários de saúde para cobrir 100% da população é de 1 a cada 750 pessoas. Tomando como base esse indicativo, temos um valor mínimo necessário de 1.3 ACS a cada grupo de mil pessoas. A figura 31 mostra o *ranking* de cidades com mais de cem mil habitantes ordenados pela quantidade de agentes a cada mil habitantes (MINISTÉRIO DO ESTADO DA SAÚDE, 2006).

Figura 31 - Indicador qualitativo de Agentes Comunitários de Saúde.




Fonte: Autoria própria.

Como é possível observar na figura 31, para cidades com população acima de 100 mil habitantes, são pouquíssimos os casos em que o valor de 1,3 ACS a cada mil habitantes é respeitado, evidenciando assim uma necessidade por parte do poder público de criação e

manutenção de cargos de agentes comunitários de saúde para garantir a qualidade de vida e saúde aos cidadãos.

Por se tratar de um *dashboard* dinâmico, além dos resultados aqui apresentados, o usuário final pode utilizá-lo de acordo com a sua necessidade específica. Ele pode, por exemplo, verificar em sua cidade se determinado profissional atende ou não pelo SUS, pesquisar seu local de atendimento (e o local mais perto caso atenda em mais de um lugar). É possível também, realizar a busca por Psicólogos ou Psicopedagogos na cidade em questão, mostrando que a ferramenta tem ampla utilização para análise de diversos cenários da área da saúde.

Todas as visualizações criadas e indicadores calculados só foram possíveis de serem realizados devido ao resultado do processo de ETL, que teve como objetivo agrupar em uma única fonte todos os dados necessários para o desenvolvimento do trabalho, possibilitando, através dessa visão, um cenário onde o poder público possa utilizar esse tipo de informação para auxílio na tomada de decisões.

## 5 CONCLUSÃO

Através do desenvolvimento deste trabalho foi possível evidenciar a importância de uma estrutura organizacional, no caso, na área de saúde do estado de São Paulo, que possibilite a geração de indicadores através dos dados do Cadastro Nacional de Estabelecimentos de Saúde (CNES) para o estado de São Paulo. A instituição deve contar com uma equipe técnica de desenvolvimento responsável por todo o fluxo de informações dessa organização, além de mostrar o ganho de tempo nas análises feitas quando se tem uma estrutura de processamento de dados por trás. É importante ressaltar também, o cenário de orientação para tomada de decisões através de uma cultura organizacional que se baseie na análise de dados e em seus modelos.

Os resultados, após o desenvolvimento do processo de ETL, possibilitaram a compreensão da distribuição tanto de profissionais da área da saúde, quanto de estabelecimentos cadastrados no CNES. Através da avaliação desses dados, foram analisados os indicadores quantitativos e qualitativos referentes a realidade da saúde pública e privada nas diferentes regiões do Estado de São Paulo, possibilitando gerar um painel interativo onde o usuário pudesse navegar por diferentes cenários e analisar as reais necessidades dos municípios paulistas, servindo de base para a tomada de decisão pelo poder público em criar incentivos que possam mitigar os cenários identificados.

Ao analisarmos a quantidade de médicos a cada mil habitantes, conclui-se que 22 das 80 cidades com mais de cem mil habitantes não atingem o valor mínimo recomendado pela OCDE de 3,5 médicos a cada mil habitantes. Para o mesmo conjunto de cidades, ao analisarmos a distribuição de agentes comunitários de saúde, apenas a cidade de Caraguatatuba ultrapassa a marca de 1,3 ACS a cada mil habitantes, valor recomendado pelo Ministério da Saúde.

Conclui-se que a ferramenta implementada se mostrou muito eficiente para que usuário possa utilizá-la de acordo com as suas necessidades. Além disso, ela pode ser utilizada pelo poder público para análise dos dados através de um dashboard dinâmico, e cujas informações podem auxiliar na tomada de decisões.

# 6 TRABALHOS FUTUROS

Como sugestão para trabalhos futuros, propõe-se uma solução que analise os dados dos diferentes Estados do Brasil, considerando também, mais de uma fonte, além da plataforma CNES, como portais públicos de acesso a dados, possibilitando a geração de mais indicadores. Para isso, seria aconselhável a estruturação de um processo que possibilite a escalabilidade, uma vez que o conjunto de dados será maior, fazendo uso, por exemplo, de armazenamento de dados em nuvem públicas como Azure e AWS, além de ferramentas para processamento de dados como *Databricks*.

**REFERÊNCIAS**


ARAVIND. **Wget Command**. [*S. l.*], 23 mar. 2022. Disponível em: https://medium.com/@aravind27032002/wget-command-55c85c82c9aa. Acesso em: 3 dez. 2022.

ASHWIN, M. **OS Module -Python**. [*S. l.*], 21 jul. 2021. Disponível em: https://medium.com/@ashes192000/os-module-python-9356b1befd00. Acesso em: 3 dez. 2022.

BOROZENETS, Michael. **Why use PostgreSQL as a Database for my Next Project in 2022**. [*S. l.*], 14 jul. 2022. Disponível em: https://fulcrum.rocks/blog/why-use-postgresql-database#why-we-use-postgresql-3. Acesso em: 5 dez. 2022.

COUTINHO, Thiago. **Como o Anaconda IDE pode ajudar na sua programação em Python**: Descubra como o Anaconda IDE pode auxiliar no desenvolvimento de códigos de ciência de dados e machine learning. [*S. l.*]: Voitto, 13 ago. 2020. Disponível em: https://www.voitto.com.br/blog/artigo/anaconda-python-ide. Acesso em: 1 dez. 2022.

DIAS, Tiago. **Manipulando Dados em PostgreSQL com Python**. [*S. l.*], 20 maio 2021. Disponível em: https://dadosaocubo.com/manipulando-dados-em-postgresql-com-python/. Acesso em: 3 dez. 2022.

GROSSI, Pedro. SP é o Estado com menor diferença no número de médicos entre Capital e interior. **Revide**, [*S. l.*], p. 1, 26 mar. 2018. Disponível em: https://www.revide.com.br/noticias/saude/sp-e-o-estado-com-menor-diferenca-no-numero-de-medicos-entre-capital-e-interior/#:~:text=Os%20dados%20apontam%20que%20a,as%20unidades%20federativas%20do%20Pa%C3%ADs. Acesso em: 6 abr. 2023.

GUIA MÉDICO. [*S. l.*], 2020. Disponível em: https://guiamedico.cremesp.org.br/. Acesso em: 5 jan. 2023.

GUIMARÃES, Juca. Por que médicos brasileiros se recusam a trabalhar no interior? **Brasil de Fato**, [*S. l.*], p. 1, 23 nov. 2018.

HENRIQUE, Pedro. Introdução ao Jupyter Notebook Para Python. Meidum, [S. l.], p. 1, 29 jun. 2020. Disponível em: https://medium.com/@pedrofullstack/introdu%C3%A7%C3%A3o-ao-jupyter-notebook-para-python-b2cf79cea31d. Acesso em: 19 abr. 2023.

MAHMOOD, Omer. What's ETL?: and why it's critical for data science. **Towards Data Science**, [*S. l.*], p. 1, 1 mar. 2021. Disponível em: https://towardsdatascience.com/whats-etl-b4903a57f8ce. Acesso em: 6 abr. 2023.

MAIA, Filipe. **O que é Power BI: para que serve e como utilizar?**. [*S. l.*], 27 out. 2021. Disponível em: https://www.leansolutions.com.br/blog/power-bi/#h-o-que-e-power-bi. Acesso em: 8 dez. 2022.

MINISTÉRIO DE ESTADO DA SAÚDE. PORTARIA Nº 648/GM DE 28 DE MARÇO DE 2006. **Política Nacional de Atenção Básica**, [*S. l.*], 28 mar. 2006.

PORTELLA, Michelle. Brasil só alcançará média de 3,5 médicos por cada 1 mil habitantes em 2030: País tem atualmente 2,8 médicos para cada 1 mil habitantes, média distante do patamar recomendado pela OCDE. **Correio Braziliense**, [*S. l.*], p. 1, 24 jan. 2023.

POZO, Leodanis. **Python's zipfile: Manipulate Your ZIP Files Efficiently**. [*S. l.*], 14 fev. 2022. Disponível em: https://realpython.com/python-zipfile/. Acesso em: 3 dez. 2022.

RAI, Abhijeet. **5 Must know uses of the 'sys' module in Python**. [*S. l.*], 1 fev. 2021. Disponível em: https://medium.com/analytics-vidhya/5-must-know-uses-of-the-sys-module-in-python-9931ab2e41e7. Acesso em: 3 dez. 2022.

REIS, Joe. **Fundamentals of Data Engineering**. 1. ed. [*S. l.*]: O'Reilly Media, 2022. 740 p.

SCHEFFER, Mario. **Demografia Médica no Brasil 2018**. [*S. l.*: *s. n.*], 2018. Disponível em: https://jornal.usp.br/wp-content/uploads/DemografiaMedica2018.pdf. Acesso em: 20 jan. 2023.

SCHEFFER, Mario. **Demografia Médica no Brasil 2023**. [*S. l.*: *s. n.*], 2023. Disponível em: https://amb.org.br/wp-content/uploads/2023/02/DemografiaMedica2023_8fev-1.pdf. Acesso em: 5 abr. 2023.

SINHA, Shubham. **Apache Spark Architecture -Distributed System Architecture Explained**. [*S. l.*]: Edureka, 28 set. 2018. Disponível em: https://medium.com/edureka/spark-architecture-4f06dcf27387#:~:text=in%20Spark%20Shell-,Spark%20%26%20its%20Features,processing%20speed%20of%20an%20application. Acesso em: 1 dez. 2022.

TAURION, Cézar. **Big Data**. 1. ed. [*S. l.*]: BRASPORT, 2013. 126 p.

TAURION, Cézar. Os 5 Vs do Big Data: As oportunidades que trazem não podem nem devem ser desperdiçadas. **Itforum**, [*S. l.*], p. 1, 17 jun. 2016. Disponível em: < https://itforum.com.br/noticias/os-5-vs-do-big-data/> Acesso em: 6 mar. 2023.

THATHA, Viswanath. **Scheduling scripts using Task Scheduler**. [*S. l.*], 31 out. 2020. Disponível em: https://medium.com/analytics-vidhya/scheduling-scripts-using-task-scheduler-ced4d98a5be5. Acesso em: 8 dez. 2022.