

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Engenharia de Dados

Gabriel Luis Rodrigues

**ANÁLISE DA PRÁTICA DE ATIVIDADE FÍSICA ENTRE ADULTOS
BRASILEIROS NAS CAPITAIS**

Belo Horizonte
2023

Gabriel Luis Rodrigues

**ANÁLISE DA PRÁTICA DE ATIVIDADE FÍSICA ENTRE ADULTOS
BRASILEIROS NAS CAPITALS**

Trabalho de Conclusão de Curso, apresentado
ao Curso de Especialização em Engenharia de
Dados como requisito parcial à obtenção do
título de especialista.

Belo Horizonte

2023

SUMÁRIO

1. Introdução.....	4
1.1. Contextualização	4
1.2. O problema proposto	5
2. Coleta de Dados	6
3. Ingestão/Processamento/Tratamento de Dados.....	9
4. Orquestração de Dados	13
5. Visualização de dados	18
6. Apresentação dos Resultados	25
7. Links	26
8. Referências.....	26
APÊNDICE.....	27

1. Introdução

1.1. Contextualização

A prática de atividade física é muito importante para o fortalecimento do corpo e também pode oferecer benefícios psicológicos, como promover a sensação de bem-estar. A redução dos níveis de atividade física e o aumento do tempo gasto em atividades sedentárias pode fazer com que a população desenvolva doenças cardiovasculares, respiratórias, metabólicas, musculoesqueléticas, câncer e depressão .

Por outro lado, a realização de atividade física é considerada como uma estratégia para a prevenção e gestão de doenças crônicas e ainda para a melhora da saúde mental, reduzindo o risco de depressão e deficiência cognitiva e melhorando e elevando a autoestima.

No Brasil, a inatividade física é responsável por 3% a 5% de todas as principais DCNT (Doenças crônicas não transmissíveis) e 5,31% de todas as causas de mortalidade, variando de 5,82% na região sudeste a 2,83% na região sul (Silva et al., 2017).

De acordo com a OMS, um em cada quatro adultos no mundo não atinge os níveis globais recomendados de atividade física, o que reflete nos cinco milhões de mortes por ano que poderiam ser evitadas se a população global fosse mais ativa (World Health Organization, 2018). O cumprimento ou não das recomendações sobre a prática da atividade física é uma variável investigada em diversos estudos e em diferentes países. Por exemplo, no Brasil, a taxa de sedentarismo aumentou desde 2002 em mais de 15%; e dados indicam que mais de 47% dos brasileiros em 2016 eram sedentários (Guthold, Stevens, Riley, & Bull, 2018).

A prática da atividade física é descrita como um dos principais fatores comportamentais de proteção das Doenças Crônicas, pela redução do risco de mortalidade por todas as causas, prevenção de doenças cardiovasculares (DCV) e diabetes, melhoria dos níveis de lipídios, redução da hipertensão e dos riscos de câncer de mama e cólon (Bull, Goenka, Lambert, & Pratt, 2017). De acordo com esses autores citados, ainda tem efeitos positivos na saúde mental, retarda o início da demência e pode ajudar na manutenção de um peso saudável.

1.2. O problema proposto

Neste trabalho, será utilizado a Ingestão Transformação e Gravação de dados para uma Análise Exploratória, para extração de informações importantes das práticas de atividade física, com o objetivo de realizar uma análise e encontrar padrões para classificá-lo “SIM” ou “Não” para prática de atividades físicas. Para isso, todos os atributos serão classificados com um grau de importância. Desta maneira, conseguimos analisar os resultados e utilizá-los em previsões futuras.

Serão analisados, os dados de prática de atividade física, disponibilizados (VIGITEL) e IBGE. Os principais objetivos dessa análise são:

- Realizar uma análise nos dados das práticas de atividade física por capitais. Desta forma, auxiliaremos para ver qual atividade é mais praticada em determinada região e idade do praticante.
- Os dados que serão analisados, foram coletados do site da (VIGITEL) e IBGE. Foi necessário coletar algumas informações separadamente, são elas:
 1. Dataset das práticas de atividade física: neste dataset são apresentadas informações sobre atividade física; contendo informações como: se o entrevistado pratica alguma atividade, tipo de atividade, altura e peso do entrevistado, tempo que se dedica a prática da atividade, região geográfica do entrevistado, bem como sua idade, gênero e escolaridade.
 2. Dataset dos valores Demográficos por Capital: neste dataset é apresentado o valor demográfico da capital, como população estimada, sigla dos estados, renda per capita, escolaridade, IDH, região etc...

- As análises realizadas, têm como objetivo encontrar padrões, métricas e tendências que auxiliarão no entendimento das bases trabalhadas. E assim, poderemos indicar quais características da atividade física praticada, gênero, tipo da prática e o tempo dedicado, para fins de entender quais são as atividades praticadas por região e o tempo da prática.

¹ <https://svs.aids.gov.br/download/Vigitel/>

² <https://cidades.ibge.gov.br/brasil/>

VIGITEL(Sistema de Vigilância de Fatores de Risco e Proteção para Doenças Crônicas por Inquérito Telefônico)

2. Coleta de Dados

Para o tratamento do problema proposto, foram utilizados cinco *datasets* (conjunto de dados). O processo amostral utilizado no VIGITEL foi do tipo probabilístico e foi constituído por sorteio de 5.000 residências com linha telefônica fixa por cidade, seguido de sorteio de um morador com idade >18 anos por domicílio até se obter o número mínimo de 2.000 entrevistas por cidade. O número de entrevistas completas realizadas pelo VIGITEL a cada ano foi: 53.210 em 2018, 53.034 em 2019, 52.395 em 2020 e 52.443 entrevistas em 2021, totalizando 211.082 no período, extraídos em 20/09/2023 do repositório de dados Secretaria de Vigilância em Saúde. E o dataset sobre informações demográficas foi extraído do repositório do IBGE na data de 20/09/2023

Os quatros *dataset*, “Vigitel-2018-peso-rake.xls”, Vigitel-2019-peso-rake.xls”, Vigitel-2020-peso-rake.xls”,e Vigitel-2021-peso-rake.xls” disponível em <https://svs.aids.gov.br/download/Vigitel/>, traz as informações sobre a prática de atividade física. Os primeiros quatro datasets possuem os seguintes campos conforme tabela a seguir. Usaremos um dicionário de dados(Dicionario-de-dados-Vigitel.xls) para entender o que cada campo representa no dataset e assim renomeamos para os campos abaixo:

Variável	Tipo	Descrição
----------	------	-----------

Idade	Numérico	Idade do entrevistado
Sexo	Texto	Masculino ou Feminino
Grau_escolaridade	Texto	Grau escolaridade: curso primário Admissão, curso ginásial ou ginásio, 1º grau ou fundamental ou supletivo de 1º grau 2º grau ou colégio ou técnico ou normal ou científico científico ou ensino médio ou supletivo de 2º grau, 3º grau ou curso superior pós-graduação (especialização, mestrado, doutorado), nunca estudou, não sabe e não quis responder
Peso	Numérico	Peso do entrevistado
Altura	Numérico	Altura do entrevistado
Prática_exercício	Texto	Sim ou Não
Tipo_exercício	Texto	Tipo da Atividade: caminhada (não vale deslocamento para trabalho), caminhada em esteira, corrida (cooper), corrida em esteira musculação, ginástica aeróbica (spinning, step, jump), hidroginástica, ginástica em geral (alongamento, pilates, ioga), natação, artes marciais e luta (jiu-jitsu, karatê, judô, boxe, muay thai, capoeira), bicicleta (inclui ergométrica), futebol/futsal, basquetebol, voleibol/futevolei tênis, dança (balé, dança de salão, dança do ventre) e outros.
Pratica_exercicio_1_vez_na_semana	Texto	Sim ou Não
Frequencia_exercicio	Texto	Quantidade de vezes praticadas durante a semana: 1 a 2 dias por semana, 3 a 4 dias por semana, 5 a 6 dias por semana e todos os dias (inclusive sábado e domingo).
Duracao_exercicio	Texto	Tempo da pratica da atividade: menos que 10 minutos, entre 10 e 19 minutos, entre 20 e 29 minutos, entre 30 e 39 minutos, entre 40 e 49 minutos, entre 50 e 59 minutos e 60 minutos ou mais
Cidade	Texto	Nome das capitais da pesquisa
Ano	Inteiro	Ano da pesquisa

Ordem	Inteiro	Ordem que a pesquisa foi registrada posição dos entrevistados inseridos na tabela.
Civil	Texto	Estado conjugal atual se a pessoa é: solteiro, casado legalmente, tem união estável há mais de seis meses, viúvo, separado ou divorciado e não quis informar

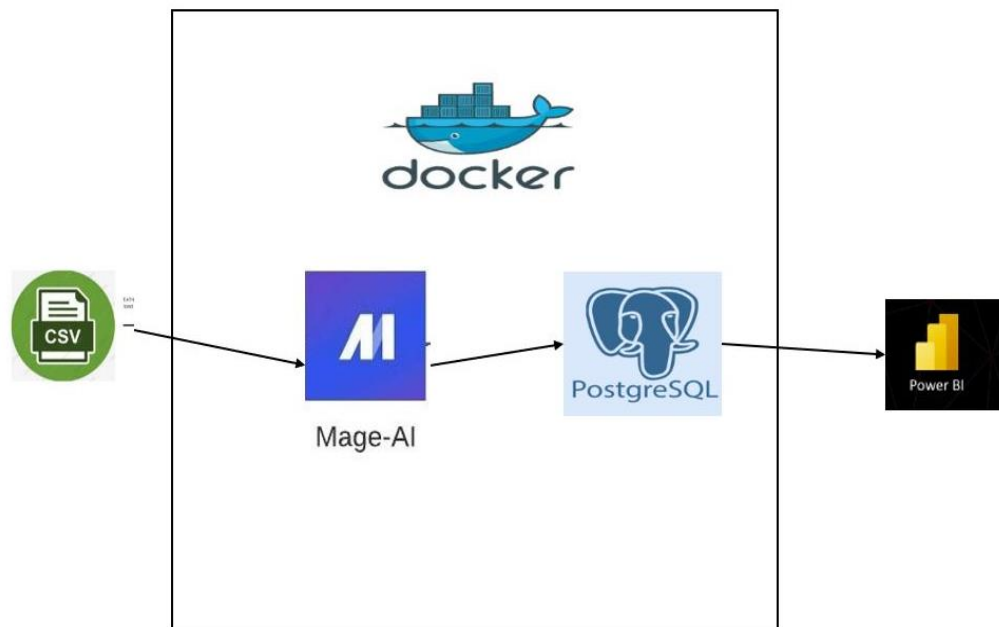
O segundo *dataset*, foi renomeado o arquivo para “Capitais_Senso.xls”, disponível em <https://www.ibge.gov.br/exportacao/08ab48cf20f15702bf692cf118ed6c3f.xls?167>, traz a informações das capitais como região, renda per capita, nome do estado sigla do estado, numero de habitantes estimada, ano pesquisa, IDH entre outras informações, vamos usar os seguintes campos:

Variável	Tipo	Descrição
Cidade	Numérico	Código da Cidade
UF	Texto	AC,AL,AP,AM,BA,CE,DF,ES,GO,MA,MT,MS, MG,PA,PB,PR,PE,PI,RJ,RN,RS,RO,RR,SC,S P,SE,TO
Capital	Texto	Nome da Capital:Rio Branco, Maceió, Macapá, Manaus, Salvador, Fortaleza, Brasília, Vitória, Goiânia, São Luís,Cuiabá, Campo Grande, Belo Horizonte, Belém,João Pessoa,Curitiba, Recife,Teresina, Rio de Janeiro, Natal,Porto Alegre, Porto Velho, Boa Vista, Florianópolis, São Paulo, Aracaju, Palmas
Populacao estimada - pessoas [2021]	Numérico	Quantidade da população estimada
Pib per capita	Numérico	Produto Interno Bruto da capitais
Regiao	Texto	Regiões do Brasil :Norte, Sul, Sudeste, Centro-Oeste e Nordeste
Ano	Numérico	Ano da pesquisa
Salario_medio_m ensal	Numérico	Salário médio mensal dos trabalhadores formais [2019]
PIB	Numérico	Produto Interno Bruto por capital valor

3. Ingestão/Processamento/Tratamento de Dados

Nessa seção será apresentado todas as ferramentas e bibliotecas utilizadas para o carregamento, processamento e o tratamento dos dados. Como ferramenta para desenvolvimento usaremos o Docker versão 4.15.0 , PostgreSQL versão 15.4, Mage versão 0.9.30 e Power BI versão 2.121.942.0. Na figura 1 temos o esboço do pipeline de dados que vamos criar.

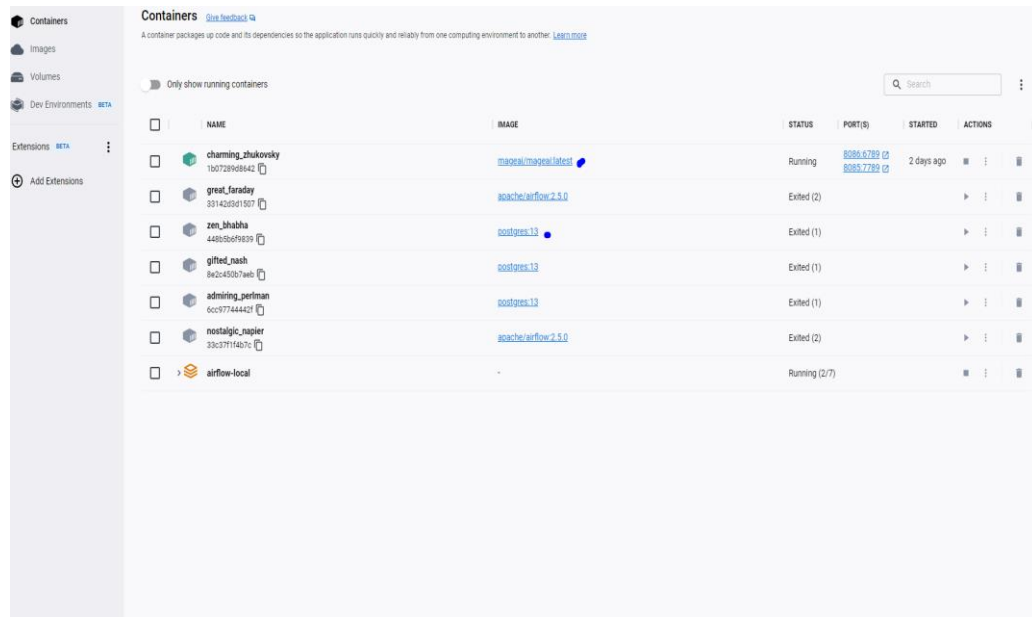
Figura 1: Screenshot do arquitetura do pipeline de dados



Fonte: Autor

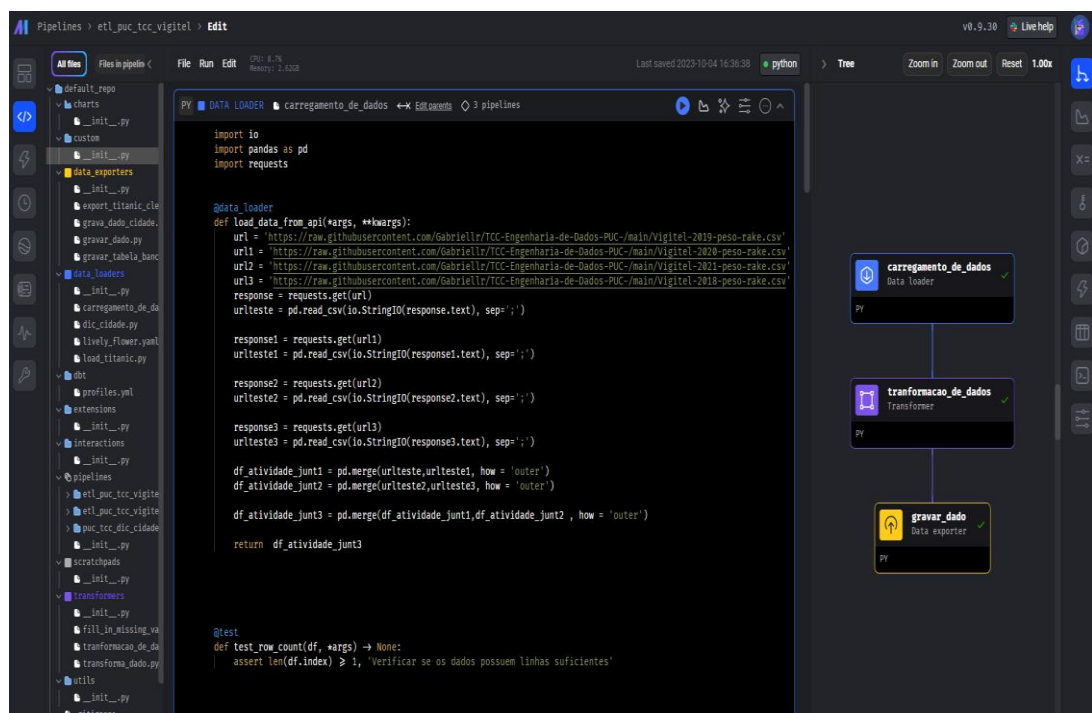
Foi escolhido o Docker para criação dos containers do Mage-AI para ingestão, processamento e tratamento dos dados(orquestração) e também o PostgreSQL para o armazenamento dos dados. Desta forma para a visualização dos dados usaremos a ferramenta Power BI. Utilizamos o Docker (figura 2) para gerir os containers.

Figura 2: Screenshot do Docker



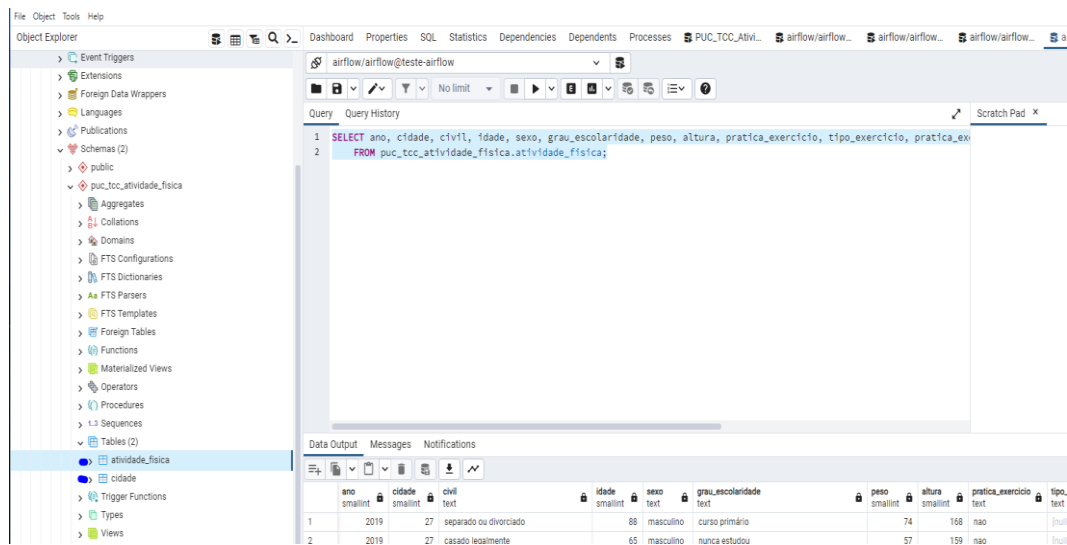
Fonte:Autor

Figura 3: Screenshot do Mage-AI



Fonte:Autor

Figura 4: Screenshot do PostgreSQL(Browse pgAdmin4)

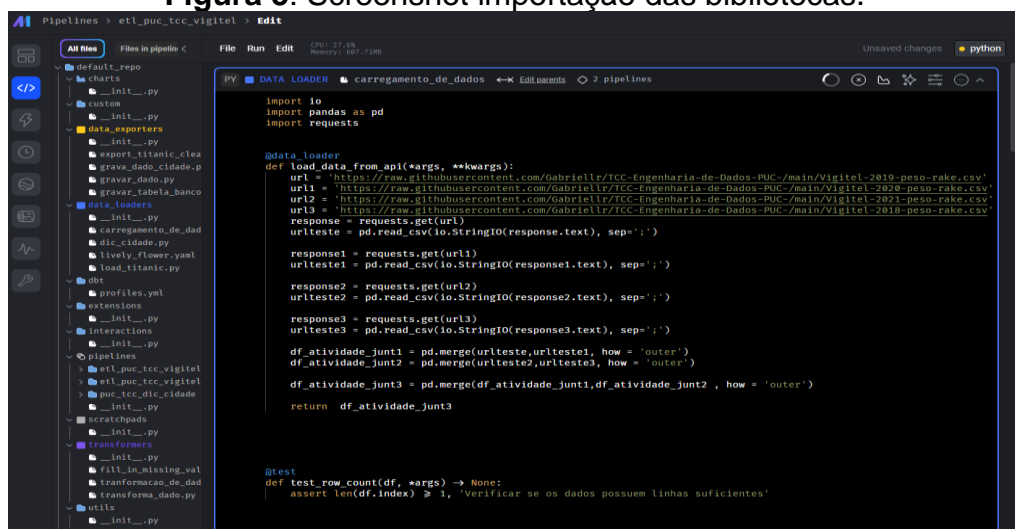


Fonte: Autor

O *Mage-IA* possui o ambiente instalado e configurado no Docker com o python3 (versão utilizada no projeto) incluso no pacote do pandas.

Para realizar a ingestão e o processamento e o tratamento dos dados, foi necessário importar algumas bibliotecas conforme a figura abaixo.

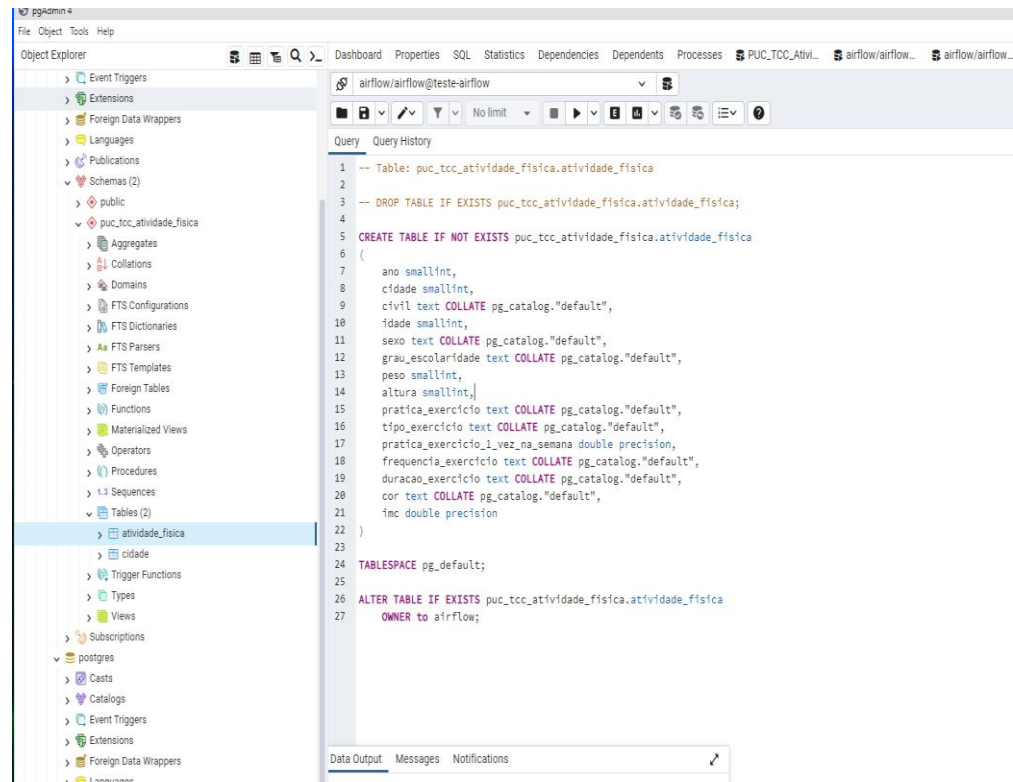
Figura 5: Screenshot importação das bibliotecas.



Fonte: Autor

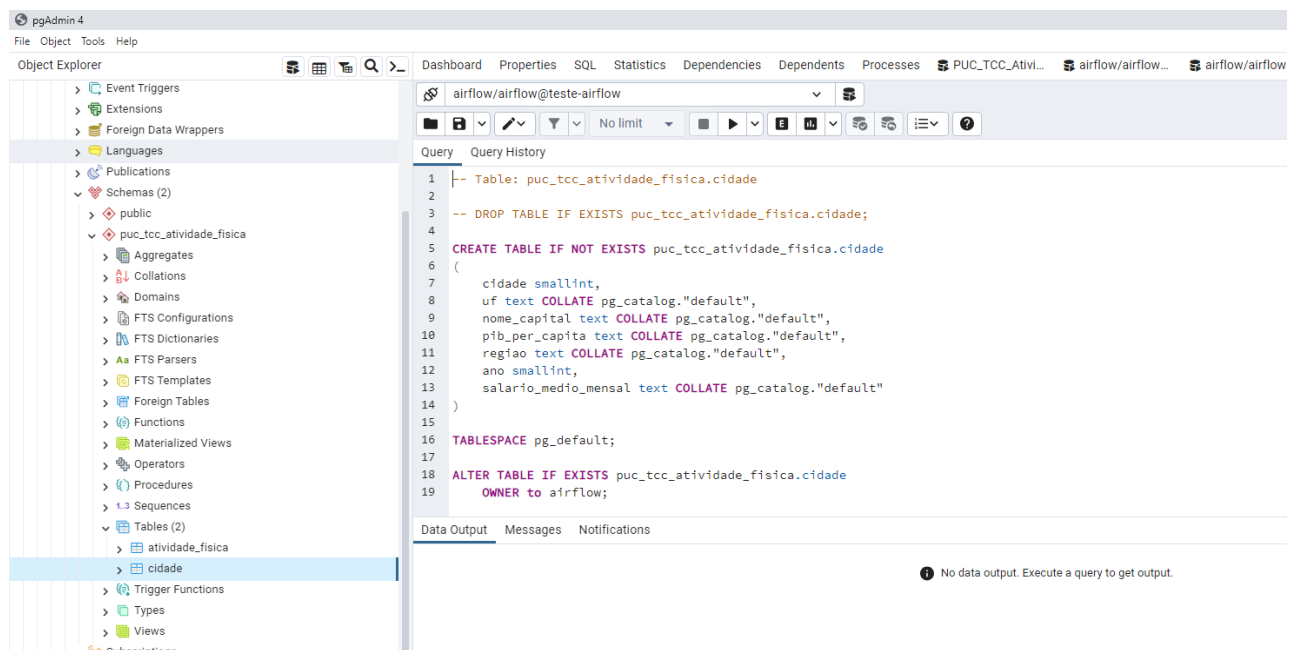
Em seguida, foi criada duas tabelas(Cidade e Atividade_Fisica) no PostgreSQL para armazenamento dos dados conforme imagem 6 e 7 abaixo.

Figura 6: Screenshot Script da tabela(Atividade Física) criada no PostgreSql.



Fonte: Autor

Figura 7: Screenshot Script da tabela(Cidade) criada no PostgreSql.

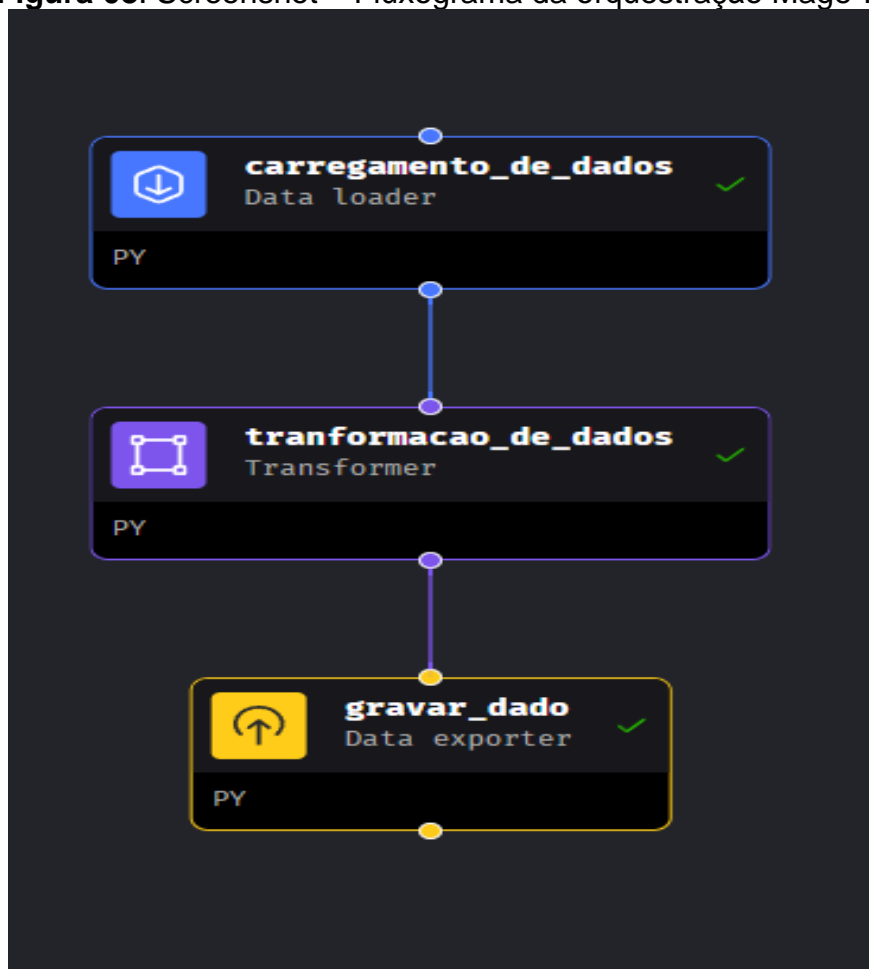


Fonte:Autor

4. Orquestração de Dados

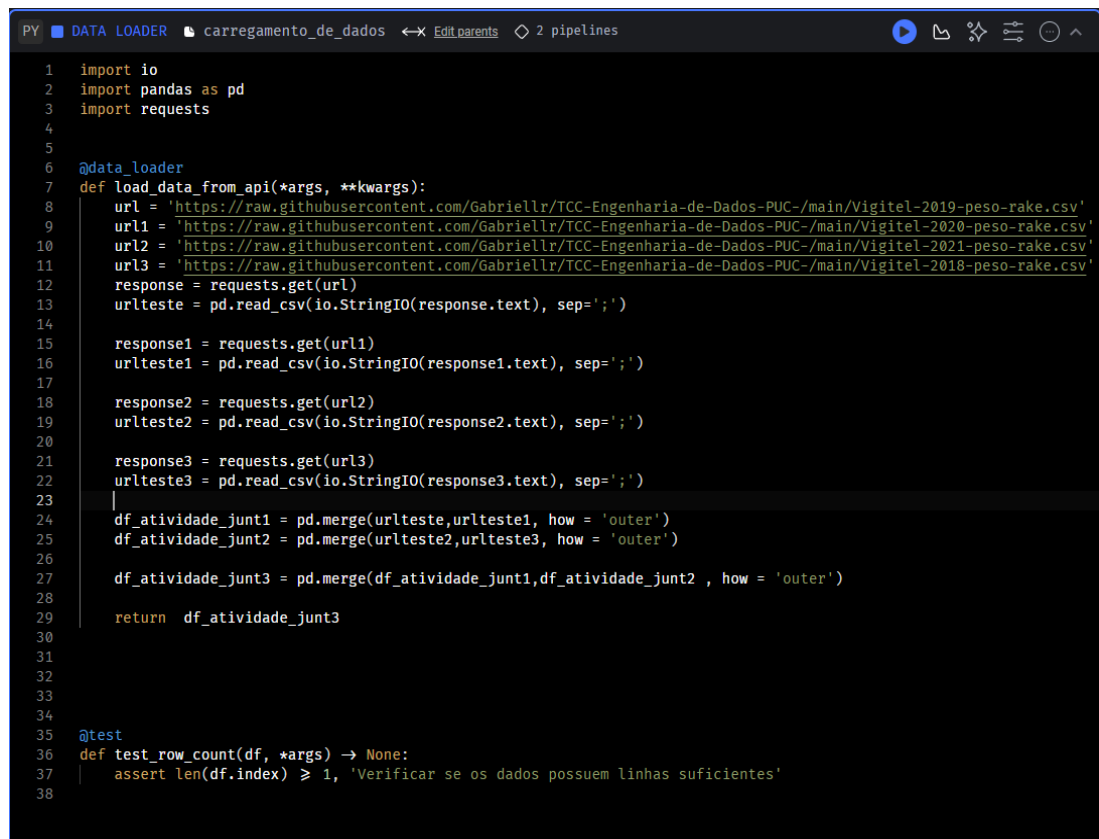
Para a realização da orquestração de dados iremos utilizar a ferramenta Mage-IA que irá coletar, transformar e gravar os dados em um banco de dados. Para coletar iremos buscar os dados através de alguns arquivos csv. Conforme imagem abaixo.

Figura 08: Screenshot – Fluxograma da orquestração Mage-IA



Fonte: Autor

Figura 9: Screenshot – Código coleta de dados Mage-IA



```

1  import io
2  import pandas as pd
3  import requests
4
5
6  @data_loader
7  def load_data_from_api(*args, **kwargs):
8      url = 'https://raw.githubusercontent.com/Gabriellr/TCC-Engenharia-de-Dados-PUC-/main/Vigitel-2019-peso-rake.csv'
9      url1 = 'https://raw.githubusercontent.com/Gabriellr/TCC-Engenharia-de-Dados-PUC-/main/Vigitel-2020-peso-rake.csv'
10     url2 = 'https://raw.githubusercontent.com/Gabriellr/TCC-Engenharia-de-Dados-PUC-/main/Vigitel-2021-peso-rake.csv'
11     url3 = 'https://raw.githubusercontent.com/Gabriellr/TCC-Engenharia-de-Dados-PUC-/main/Vigitel-2018-peso-rake.csv'
12     response = requests.get(url)
13     urlteste = pd.read_csv(io.StringIO(response.text), sep=';')
14
15     response1 = requests.get(url1)
16     urlteste1 = pd.read_csv(io.StringIO(response1.text), sep=';')
17
18     response2 = requests.get(url2)
19     urlteste2 = pd.read_csv(io.StringIO(response2.text), sep=';')
20
21     response3 = requests.get(url3)
22     urlteste3 = pd.read_csv(io.StringIO(response3.text), sep=';')
23
24     df_atividade_junt1 = pd.merge(urlteste,urlteste1, how = 'outer')
25     df_atividade_junt2 = pd.merge(urlteste2,urlteste3, how = 'outer')
26
27     df_atividade_junt3 = pd.merge(df_atividade_junt1,df_atividade_junt2 , how = 'outer')
28
29     return df_atividade_junt3
30
31
32
33
34
35  @test
36  def test_row_count(df, *args) → None:
37      assert len(df.index) ≥ 1, 'Verificar se os dados possuem linhas suficientes'
38

```

Fonte: Autor

Nessa etapa de transformação e limpeza de dados iremos usar o Python para fazer a limpeza e tratamento de dados para que possamos remover alguns dados que não são necessário para análise futura. Dentro do Mage-AI tem a possibilidade de usar diversas linguagem para tratamento de dados. Conforme a imagem abaixo podemos observar esse tratamento.

Figura 10: Screenshot – Código Transformação de dados Mage-IA

```

PY TRANSFORMER ▸ transformacao_de_dados ← 1 parent 2 pipelines
1 def number_of_rows_per_key(df, key, column_name):
2     data = df.groupby(key)[key].agg(['count'])
3     data.columns = [column_name]
4     return data
5
6
7 def clean_column(column_name):
8     return column_name.lower().replace(' ', '_')
9
10
11 @transformer
12 def transform(df, *args, **kwargs):
13
14     dados2 = df
15     ColunasSelecionada = ['orden', 'ano', 'cidade', 'civil', 'q6', 'q7', 'q8a', 'q9', 'q11', 'q42', 'q43a', 'q44', 'q45', 'q46', 'q69']
16     df_atividadeSelecionadas = dados2.filter(items=ColunasSelecionada)
17     df_atividadeSelecionadas.rename(columns={'q6':'idade', 'q7':'sexo', 'q8a':'grau_escolaridade', 'q9':'peso', 'q11':'altura', 'q42':'pratica_exercicio', 'q43a':'tipo_exercicio'})
18     df_atividadeSelecionadas['imc'] = df_atividadeSelecionadas.apply(
19         lambda row: round(row.peso / ((row.altura * row.altura)/1000)*0.1,0), axis=1)
20     df_atividadeSelecionadas['pratica_exercicio'].replace(1,'sim', inplace=True)
21     df_atividadeSelecionadas['pratica_exercicio'].replace(2,'nao', inplace=True)
22
23
24     df_atividadeSelecionadas['tipo_exercicio'].replace(1,'caminhada (não vale deslocamento para trabalho)', inplace=True)
25     df_atividadeSelecionadas['tipo_exercicio'].replace(2,'caminhada em esteira', inplace=True)
26     df_atividadeSelecionadas['tipo_exercicio'].replace(3,'corrida (cooper)', inplace=True)
27     df_atividadeSelecionadas['tipo_exercicio'].replace(4,'corrida em esteira', inplace=True)
28     df_atividadeSelecionadas['tipo_exercicio'].replace(5,'musculação', inplace=True)
29     df_atividadeSelecionadas['tipo_exercicio'].replace(6,'ginástica aeróbica (spinning, step, jump)', inplace=True)
30     df_atividadeSelecionadas['tipo_exercicio'].replace(7,'hidroginástica', inplace=True)
31     df_atividadeSelecionadas['tipo_exercicio'].replace(8,'ginástica em geral (alongamento, pilates, ioga)', inplace=True)
32     df_atividadeSelecionadas['tipo_exercicio'].replace(9,'natação', inplace=True)
33     df_atividadeSelecionadas['tipo_exercicio'].replace(10,'artes marciais e luta (jiu-jitsu, karatê, judô, boxe, muay thai, capoeira)', inplace=True)
34     df_atividadeSelecionadas['tipo_exercicio'].replace(11,'bicicleta (inclui ergométrica)', inplace=True)
35     df_atividadeSelecionadas['tipo_exercicio'].replace(12,'futebol/futsal', inplace=True)
36     df_atividadeSelecionadas['tipo_exercicio'].replace(13,'basquetebol', inplace=True)
37     df_atividadeSelecionadas['tipo_exercicio'].replace(14,'voleibol/futevolei', inplace=True)
38     df_atividadeSelecionadas['tipo_exercicio'].replace(15,'tênis', inplace=True)
39     df_atividadeSelecionadas['tipo_exercicio'].replace(16,'dança (balé, dança de salão, dança do ventre)', inplace=True)
40     df_atividadeSelecionadas['tipo_exercicio'].replace(17,'outros', inplace=True)
41     df_atividadeSelecionadas['grau_escolaridade'].replace(1,'curso primário', inplace=True)
42     df_atividadeSelecionadas['grau_escolaridade'].replace(2,'admissão', inplace=True)

```

Fonte: Autor

Na etapa de gravação iremos utilizar o PostgreSQL para armazenar o dados tratados iremos criar duas tabelas para que esse dado seja armazenados. Vamos utilizar uma etapa do Mage-IA para a gravação dos dados conforme imagem abaixo.

Figura 11: Screenshot – Código Gravação de dados Mage-IA

```

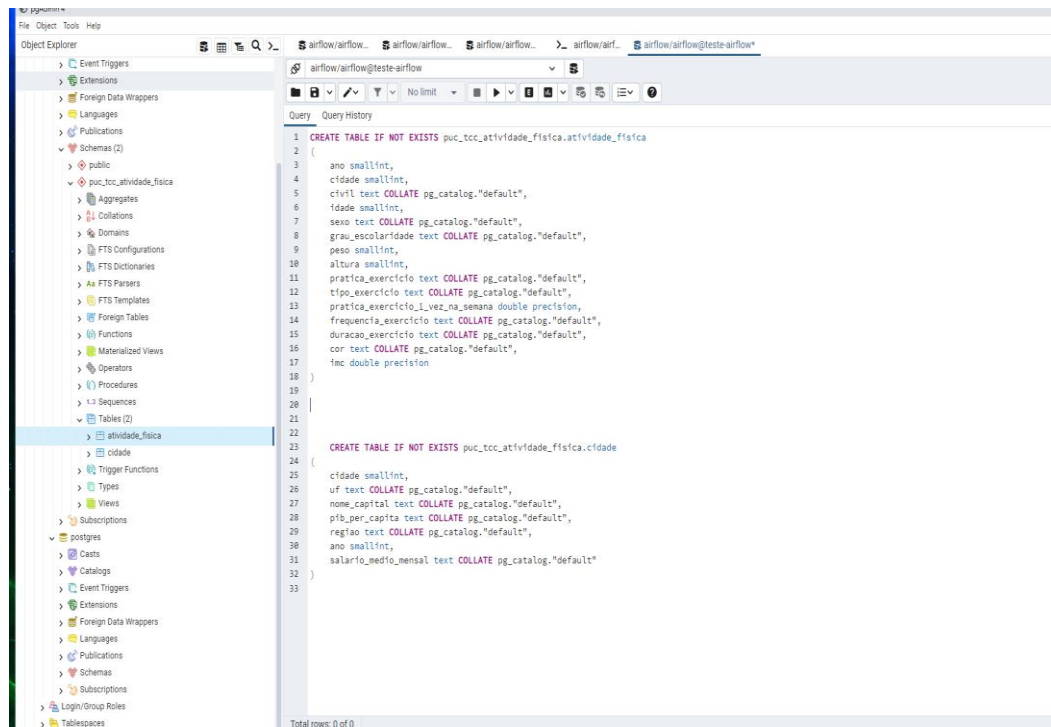
PY DATA EXPORTER ▸ gravar_dado ← 1 parent 2 pipelines
Positional arguments for decorated function:
@data_exporter
def export_data(data):
    data = transformacao_de_dados

1 from mage_ai.settings.repo import get_repo_path
2 from mage_ai.io.config import ConfigFileLoader
3 from mage_ai.io.postgres import Postgres
4 from pandas import DataFrame
5 from os import path
6
7 if 'data_exporter' not in globals():
8     from mage_ai.data_preparation.decorators import data_exporter
9
10
11 @data_exporter
12 def export_data_to_postgres(df: DataFrame, **kwargs) → None:
13     """
14     Template for exporting data to a PostgreSQL database.
15     Specify your configuration settings in 'io_config.yaml'.
16
17     Docs: https://docs.mage.ai/design/data-loading#postgresql
18     """
19     schema_name = 'PUC_TCC Atividade_Fisica' # Specify the name of the schema to export data to
20     table_name = 'Atividade_Fisica' # Specify the name of the table to export data to
21     config_path = path.join(get_repo_path(), 'io_config.yaml')
22     config_profile = 'default'
23
24     with Postgres.with_config(ConfigFileLoader(config_path, config_profile)) as loader:
25         loader.export(
26             df,
27             schema_name,
28             table_name,
29             index=False, # Specifies whether to include index in exported table
30             if_exists='replace', # Specify resolution policy if table name already exists
31         )
32

```

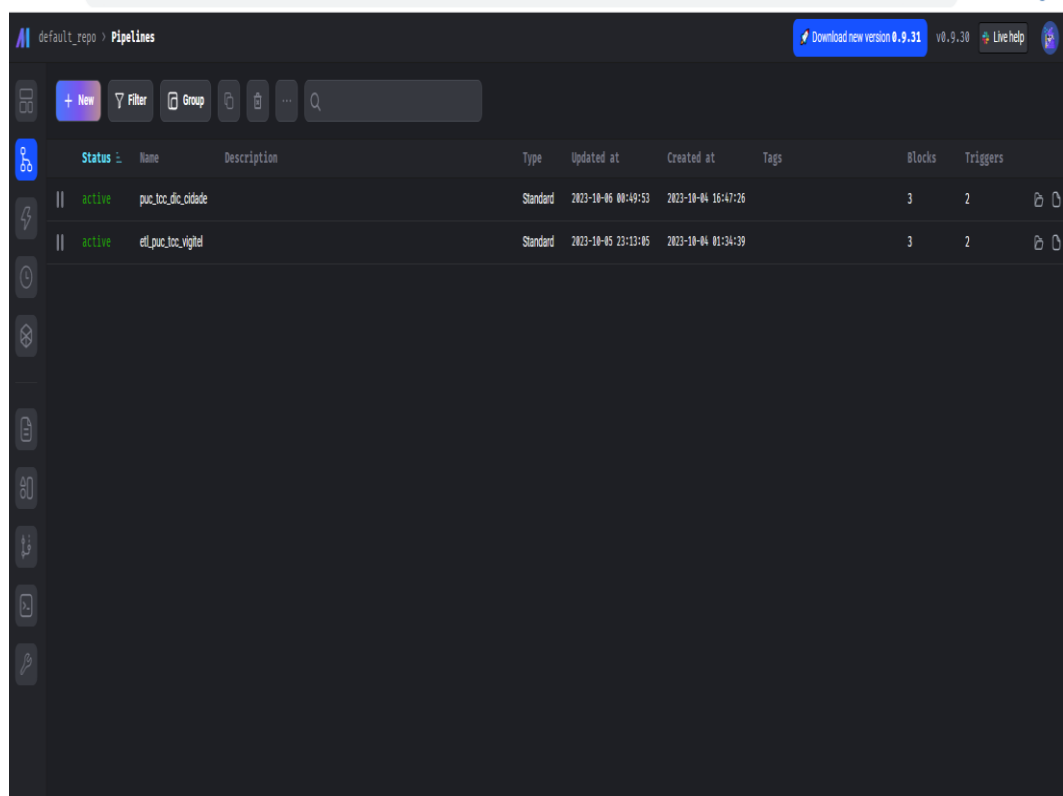
Fonte: Autor

Figura 12: Screenshot – Código Criação das tabelas PostgreSQL



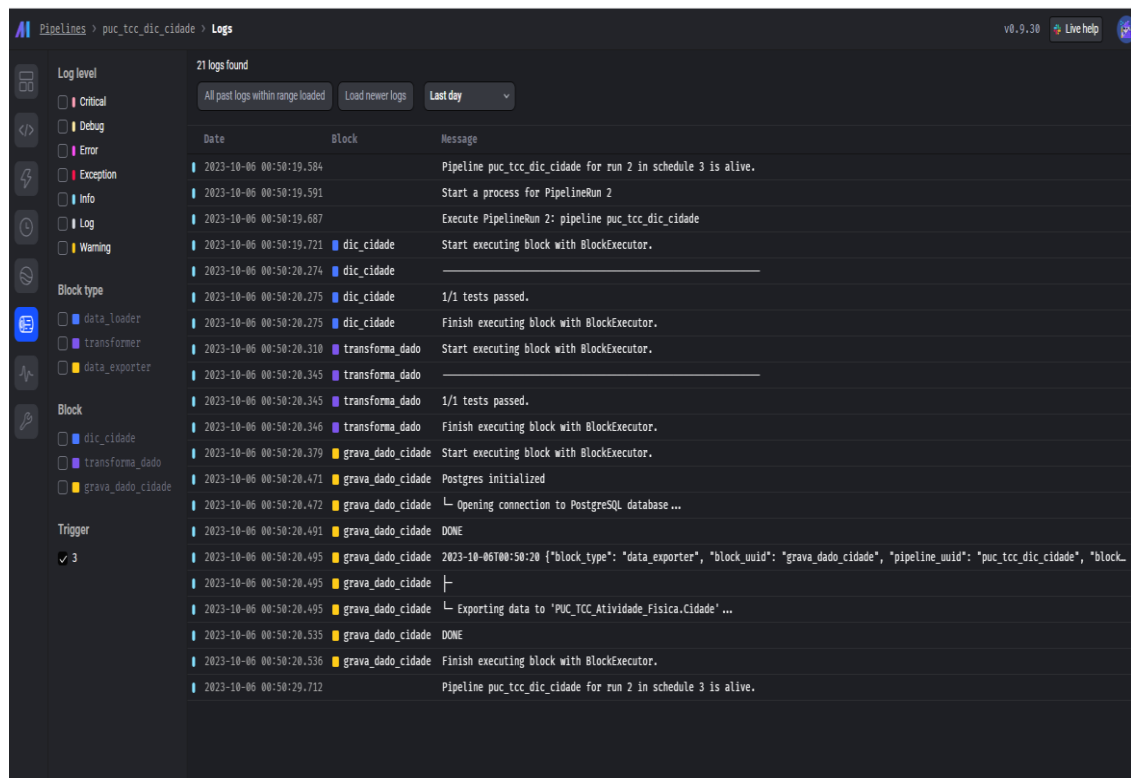
Fonte: Autor

Figura 13: Screenshot – Orquestração usando Mage-IA



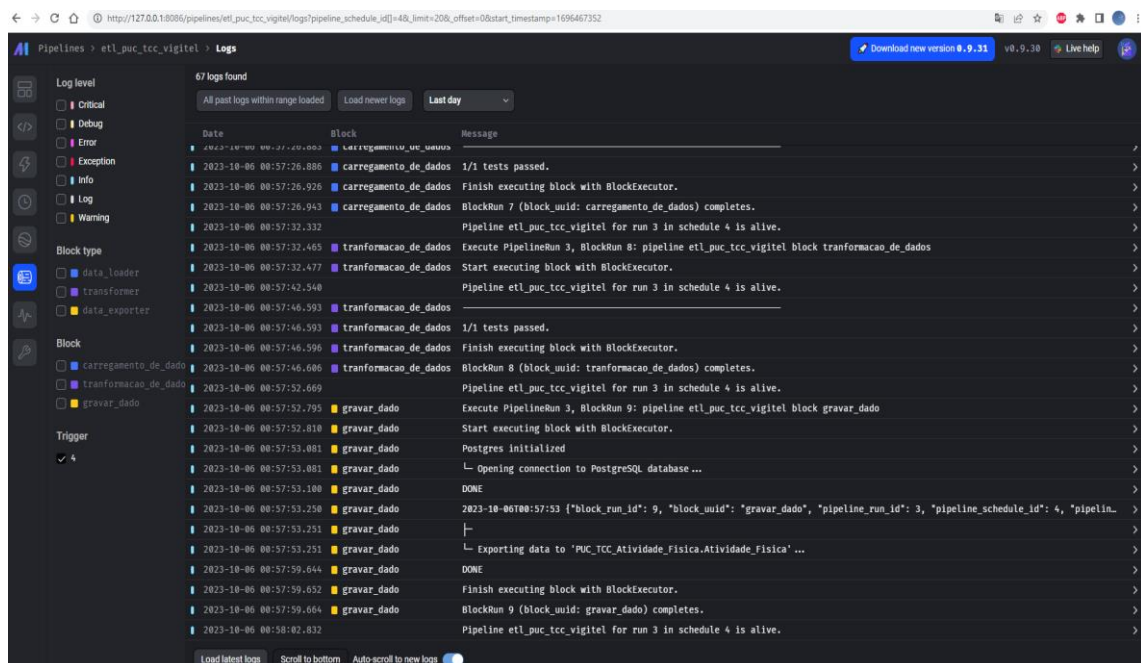
Fonte: Autor

Figura 14: Screenshot – Fluxo execução pipeline Cidade



Fonte: Autor

Figura 15: Screenshot – Fluxo execução pipeline Atividade Física

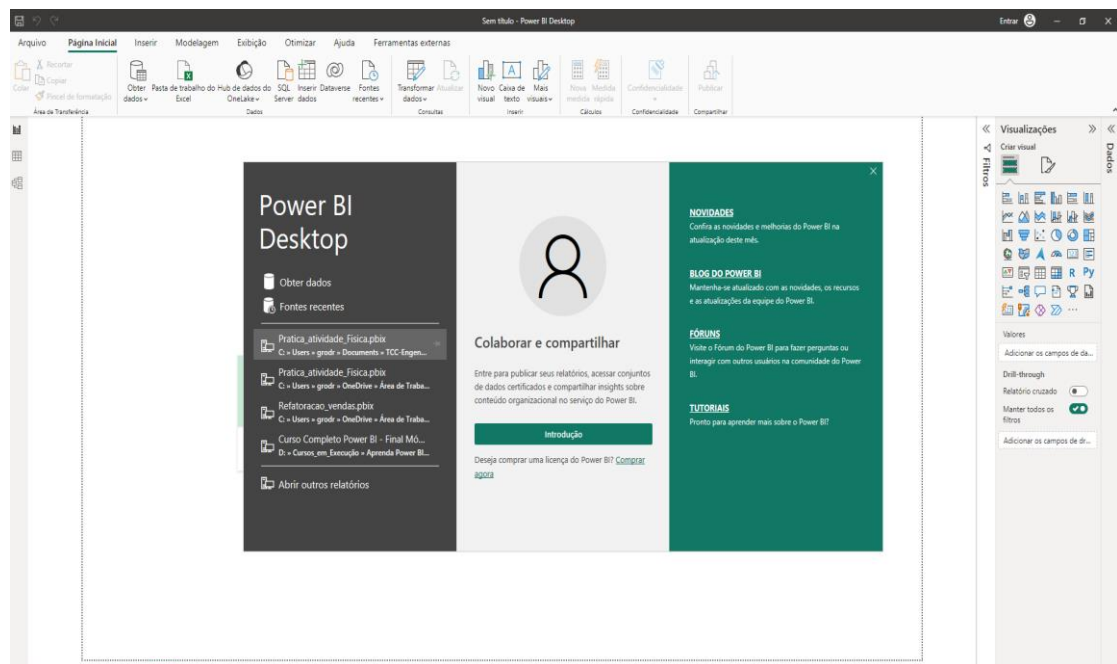


Fonte: Autor

5. Visualização de dados

Após a criação da orquestração de dados vamos criar a visualização dos dados para isso iremos utilizar a ferramenta Power BI Desktop. Conforme imagem abaixo.

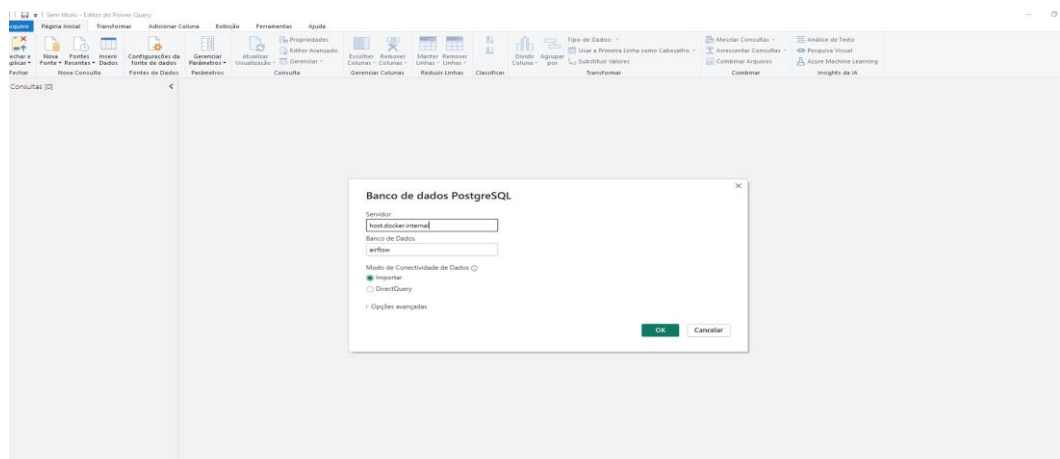
Figura 16: Screenshot – Fluxo execução pipeline Cidade



Fonte: Autor

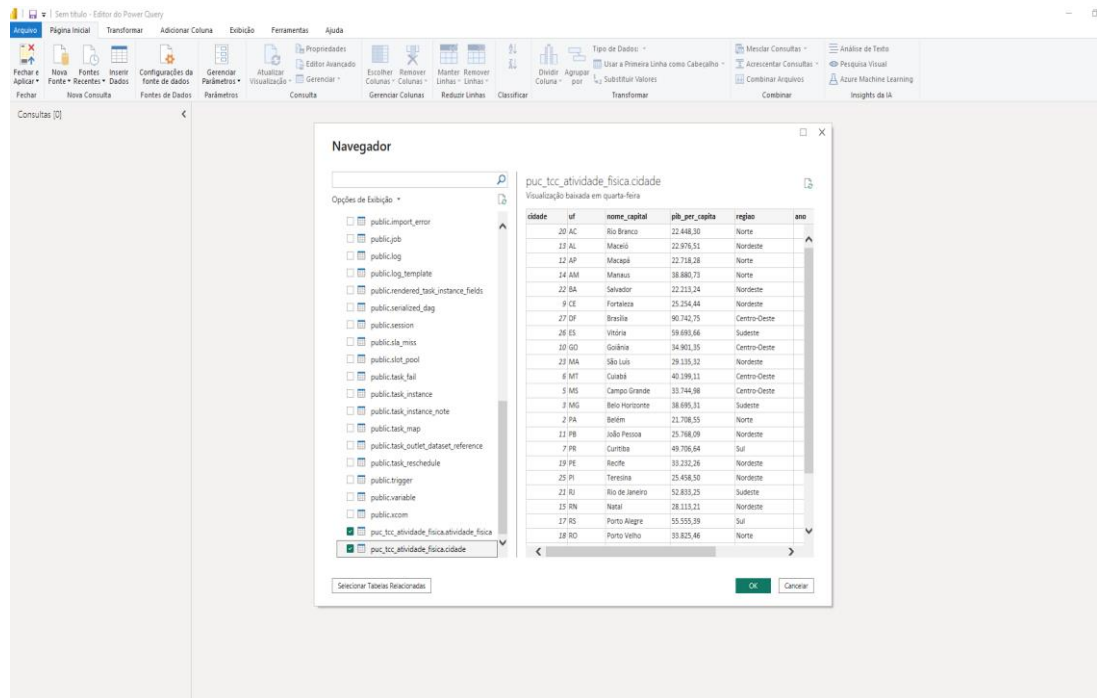
Iremos importar os dados que foram gravados no PostgreSQL usando a ferramenta Power através de uma String de conexão conectado no host local onde foi criado o banco de dados. Conforme imagem abaixo.

Figura 17: Screenshot – String conexão Power Bi PostgreSQL



Fonte: Autor

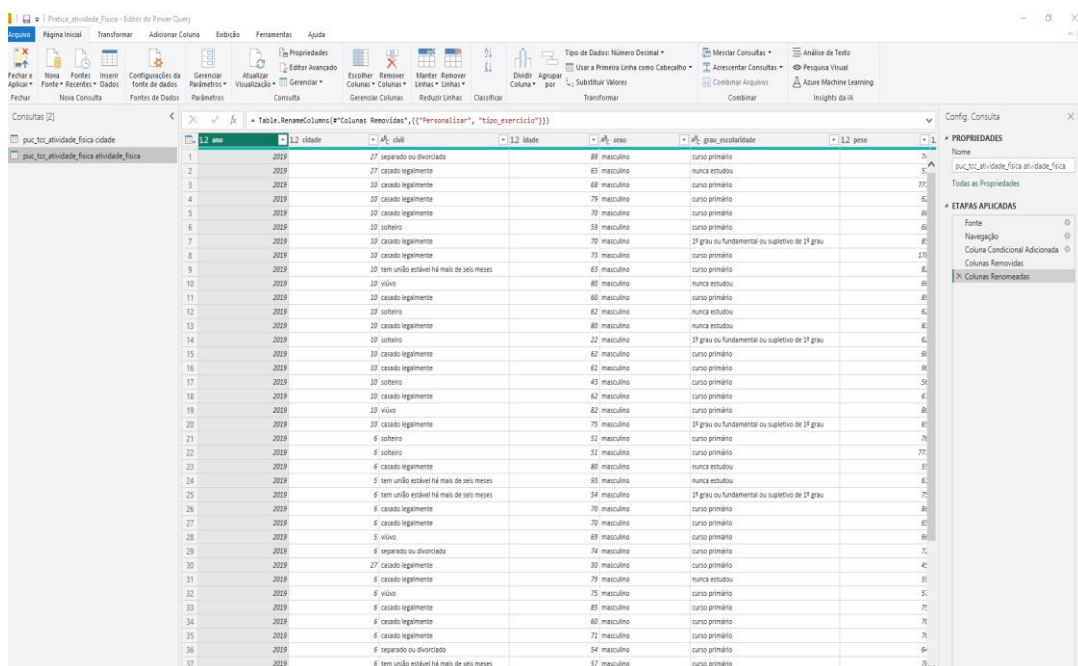
Figura 18: Screenshot – Seleção das tabelas que serão usadas



Fonte: Autor

Depois de importados os dados para dentro do Power BI, iremos fazer alguns ajustes para melhor a compreensão. Conforme imagem abaixo.

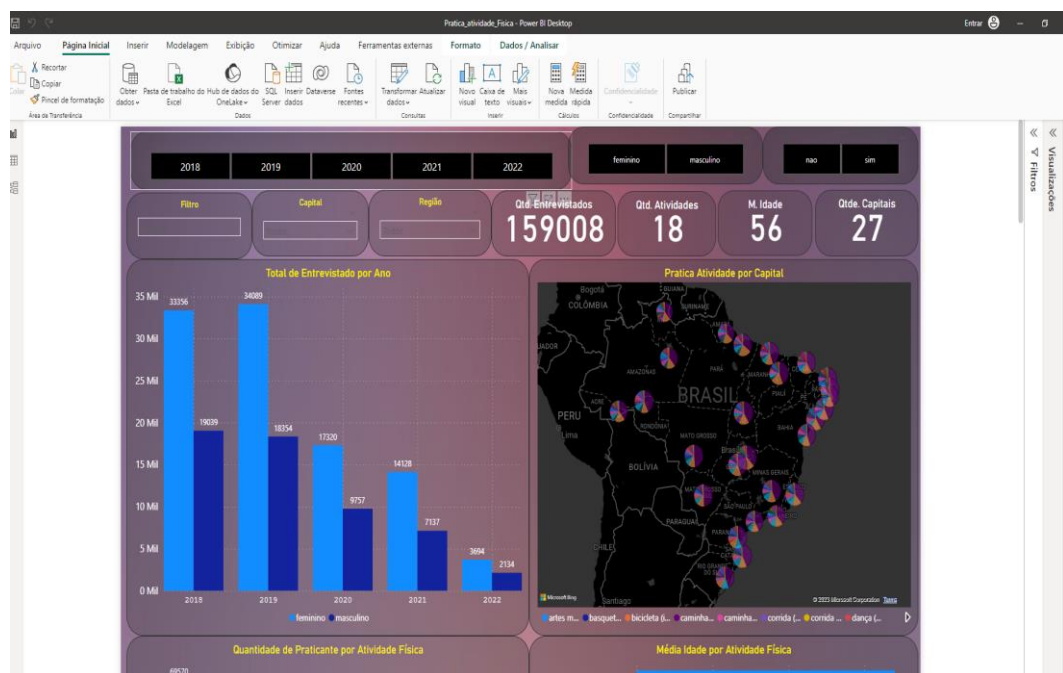
Figura 18: Screenshot – Tratamento dados Power Query



Fonte: Autor

Nesse Dashboard foram criados alguns filtros por exemplo filtro de “Datas”, “Capital”, “Região” e “Pratica atividade Sim ou Não”. Criamos alguns painel para indicar o número de entrevistados a quantidade de capitais que foram feitas as entrevistas, criamos também vários painéis para que possamos analisar melhor esse dados conforme figura abaixo.

Figura 19: Screenshot – Dashboard gerado Power BI



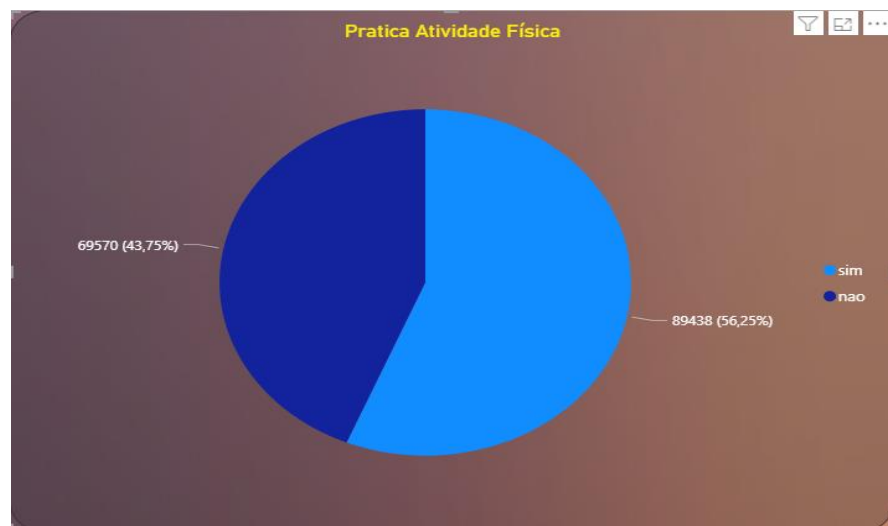
Fonte: Autor

Inicialmente, será feita a análise dos dados da tabela “atividade_fisica” de maneira individual para cada ano e, em seguida, as junções necessárias para continuação dos estudos. Portanto, as análises serão iniciadas pela tabela “atividade_fisica” que traz as informações sobre a entrevistas realizadas para identificar o número de praticante de atividade física.

Como um dos objetivos é avaliar a discrepância de característica entre os praticantes de atividade física com os não praticantes, usaremos um Data Frame contendo dados para o ano de 2018, 2019, 2020, 2021 e 2022 e depois vamos fazer uma análise descritiva mais profunda para tentar identificar alguns padrões.

Podemos verificar que mais da metade dos entrevistados praticam atividades físicas, sendo 89.438(56,25%) praticante de algum tipo de atividade física e 69.570(43,75%) que não praticam nenhum tipo de atividade. Para gerar os gráficos com seus títulos e porcentagens foram utilizados os comandos na Figura abaixo:

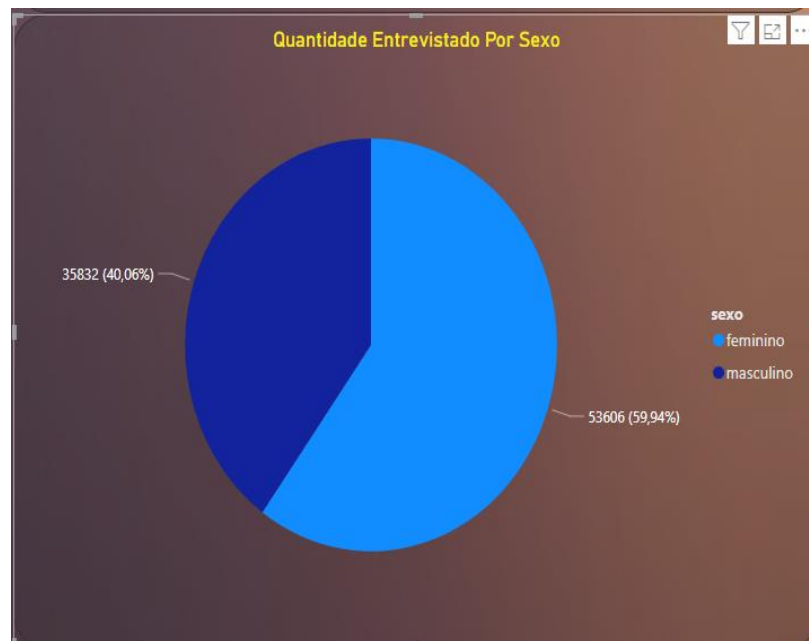
Figura 20: Screenshot – Pratica Sim ou Não



Fonte: Autor

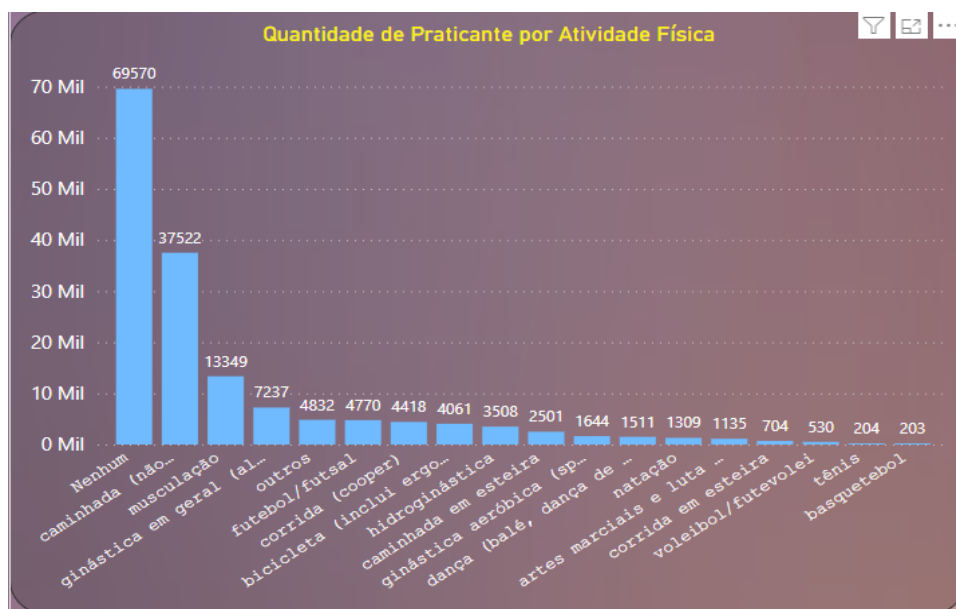
Analisando os gráficos pode-se perceber que o número de entrevistados de cada gênero já é proporcional a quantidade de praticante de atividade física. Apesar dos praticantes do sexo masculino serem a porcentagem de 43,75%, o percentual de entrevistados desse mesmo gênero é de 40,6% que são praticantes de atividade física, e 56,25% total de entrevistados do sexo feminino, sendo 59,94% praticantes de atividades física. Podemos observar na Figura abaixo.

Figura 21: Screenshot – Gênero



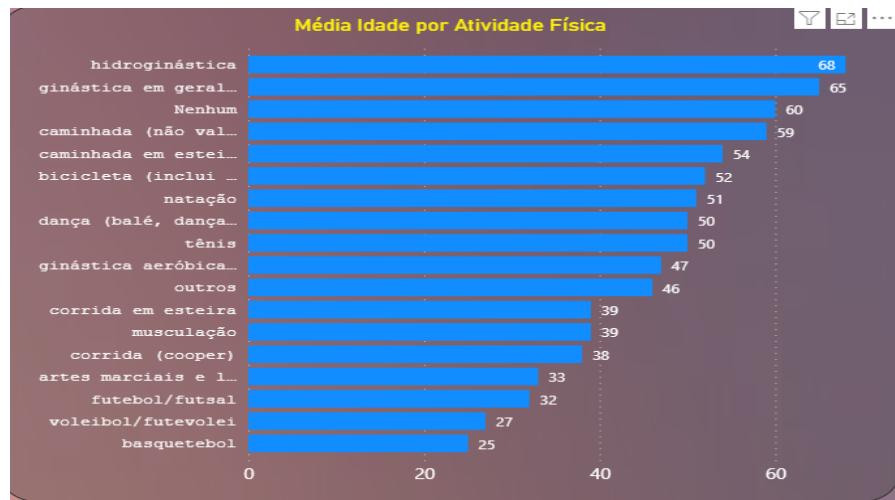
Fonte: Autor

Figura 22: Screenshot – Modalidade Atividade Física



Fonte: Autor

Figura 22: Screenshot – Média Idade Praticante de Atividade Física



Fonte: Autor

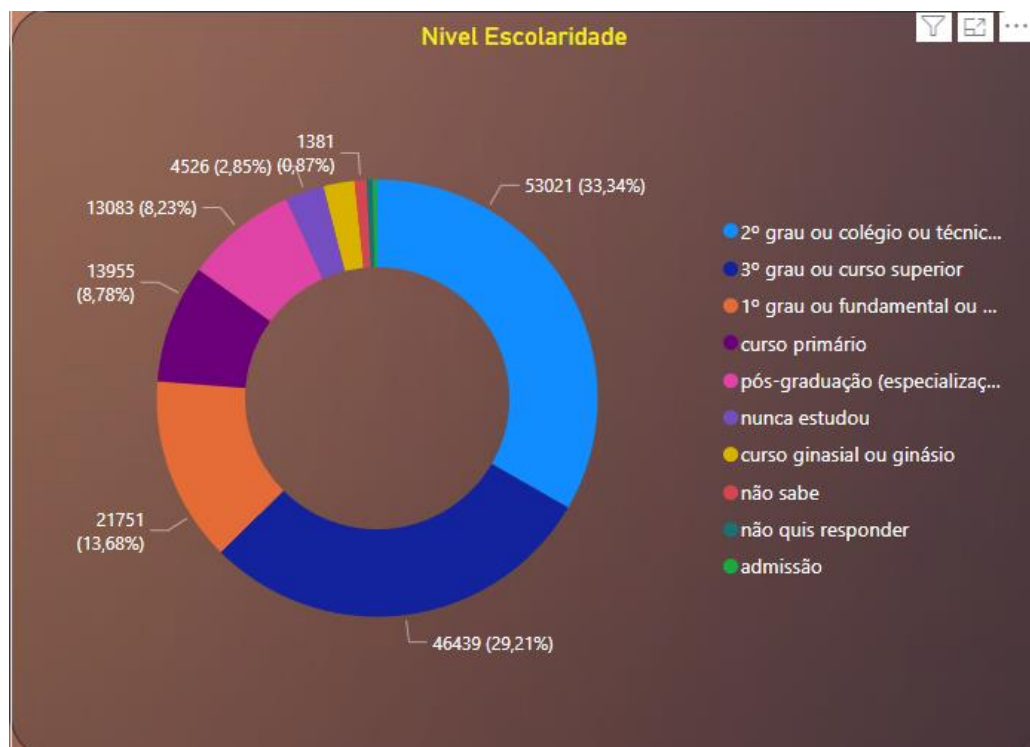
Figura 23: Screenshot – Entrevistados total agrupados por cor.



Fonte: Autor

Figura 24: Screenshot – Frequencia Atividade Física.

Fonte: Autor

Figura 25: Screenshot – Escolaridade Praticante de Atividade Física

Fonte: Autor

6. Apresentação dos Resultados

Os resultados evidenciados neste estudo corroboram importantes discussões no que se refere à criação de estratégias e condições apropriadas, não só para aumentar a frequência da prática de atividade física no lazer na população brasileira de uma forma geral, mas especialmente para a população que apresenta menor frequência desta prática, como os homens, os jovens, as pessoas de baixa escolaridade e condições socioeconômicas precárias.

Esta reflexão deve perpassar pelos profissionais de saúde, gestores públicos e sociedade civil organizada. Nesse sentido, programas populacionais e políticas de promoção da saúde voltados para incentivar a prática de atividade física na comunidade e nas escolas, assim como, políticas públicas na área do planejamento urbano e ambiente, são estratégias potencializadoras para aumentar os níveis de atividade física na população e para promoção de hábitos de vida mais saudáveis.

7. Links

Link para o vídeo:

<https://www.youtube.com/watch?v=yCv5CajRCLg>

Link relatório publicado: [Dashboard_Pratica_Atividade_Fisica](#)

Link para o repositório Github (sem datasets):

<https://github.com/Gabriellr/TCC-Engenharia-de-Dados-PUC->

8. Referencias

PYTHON. Disponível em: <<https://www.python.org/>>.

Mage-IA. Disponível em: < <https://www.mage.ai/>>.

PostgreSQL Disponível em: < <https://www.postgresql.org/>>.

DOCKER Disponível em: < <https://www.docker.com/>>.

Power BI Disponível em: < <https://powerbi.microsoft.com/>>.

World Health Organization (2018). **Global action plan on physical activity 2018–2030: more active people for a healthier world.**

<https://www.who.int/publications/i/item/9789241514187>. ISBN 978-92-4-151418-7.

Bull, F., Goenka, S., Lambert, V. & Pratt, M. (2017). **Physical Activity for the Prevention of Cardiometabolic Disease.** In: Prabhakaran D, Anand S, Gaziano TA, Mbanya JC, Wu Y, Nugent R, editors. Cardiovascular, Respiratory, and Related Disorders. (3a ed.) Washington (DC): The International Bank for Reconstruction and Development / The World Bank. 25. 10.1596/978-1-4648-0518-9_ch5

Guthold, R., Stevens, G. A., Riley, L. M., & Bull, F. C. (2018). **Worldwide trends in insufficient physical activity from 2001 to 2016:** a pooled analysis of 358 population-based surveys with 1.9 million participants. Lancet Global Health. 6(10),1077-1086. [https://doi.org/10.1016/S2214-109X\(18\)30357-7](https://doi.org/10.1016/S2214-109X(18)30357-7)

Silva, C. L., Souza, M. F., Rossi Filho, S., Silva, L. F., & Rigoni, A. C. C. (2017). **Atividade física de lazer e saúde: uma revisão sistemática. Mudanças – Psicologia da Saúde.** 25(1),57-65. <https://doi.org/10.15603/2176-1019/mud.v25n1p57-5>

APÊNDICE

Orquestração/Scripts

#Carregamento Dados Atividade Física

```
import io
import pandas as pd
import requests

@data_loader
def load_data_from_api(*args, **kwargs):
    url = 'https://raw.githubusercontent.com/Gabriellr/TCC-Engenharia-de-Dados-PUC-/main/Vigitel-2019-
    peso-rake.csv'
    url1 = 'https://raw.githubusercontent.com/Gabriellr/TCC-Engenharia-de-Dados-PUC-/main/Vigitel-2020-
    peso-rake.csv'
    url2 = 'https://raw.githubusercontent.com/Gabriellr/TCC-Engenharia-de-Dados-PUC-/main/Vigitel-2021-
    peso-rake.csv'
    url3 = 'https://raw.githubusercontent.com/Gabriellr/TCC-Engenharia-de-Dados-PUC-/main/Vigitel-2018-
    peso-rake.csv'
    response = requests.get(url)
    urlteste = pd.read_csv(io.StringIO(response.text), sep=';')
    response1 = requests.get(url1)
    urlteste1 = pd.read_csv(io.StringIO(response1.text), sep=';')
    response2 = requests.get(url2)
    urlteste2 = pd.read_csv(io.StringIO(response2.text), sep=';')
    response3 = requests.get(url3)
    urlteste3 = pd.read_csv(io.StringIO(response3.text), sep=';')
    df_atividade_junt1 = pd.merge(urlteste, urlteste1, how = 'outer')
    df_atividade_junt2 = pd.merge(urlteste2, urlteste3, how = 'outer')
    df_atividade_junt3 = pd.merge(df_atividade_junt1, df_atividade_junt2, how = 'outer')
    return df_atividade_junt3

@test
def test_row_count(df, *args) -> None:
    assert len(df.index) >= 1, 'Verificar se os dados possuem linhas suficientes'
```

#Transformação Dados Atividade Física

```
def number_of_rows_per_key(df, key, column_name):
    data = df.groupby(key)[key].agg(['count'])
    data.columns = [column_name]
    return data

def clean_column(column_name):
    return column_name.lower().replace(' ', '_')

@transformer
def transform(df, *args, **kwargs):
    dados2 = df
```

```

ColunasSelecionada = ['ordem', 'ano', 'cidade', 'civil', 'q6', 'q7', 'q8a', 'q9', 'q11', 'q42', 'q43a', 'q44',
'q45', 'q46', 'q69']
df_atividadeSelecionadas = dados2.filter(items=ColunasSelecionada )
df_atividadeSelecionadas.rename(columns={'q6': 'idade', 'q7': 'sexo', 'q8a': 'grau_escolaridade', 'q9': 'peso', 'q11':
altura', 'q42': 'pratica_exercicio', 'q43a': 'tipo_exercicio', 'q44': 'pratica_exercicio_1_vez_na_semana',
'q45': 'frequencia_exercicio', 'q46': 'duracao_exercicio', 'q69': 'cor'}, inplace = True)
df_atividadeSelecionadas['imc'] = df_atividadeSelecionadas.apply(
    lambda row: round(row.peso / (((row.altura * row.altura)/1000)*0.1), 0), axis=1)
df_atividadeSelecionadas['pratica_exercicio'].replace(1, 'sim', inplace = True)
df_atividadeSelecionadas['pratica_exercicio'].replace(2, 'nao', inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(1, 'caminhada (não vale deslocamento para trabalho)',
inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(2, 'caminhada em esteira', inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(3, 'corrida (cooper)', inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(4, 'corrida em esteira', inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(5, 'musculação', inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(6, 'ginástica aeróbica (spinning, step, jump)', inplace
= True)
df_atividadeSelecionadas['tipo_exercicio'].replace(7, 'hidroginástica', inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(8, 'ginástica em geral (alongamento, pilates, ioga)',
inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(9, 'natação', inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(10, 'artes marciais e luta (jiu-jitsu, karatê, judô, boxe,
muay thai, capoeira)', inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(11, 'bicicleta (inclui ergométrica)', inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(12, 'futebol/futsal', inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(13, 'basquetebol', inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(14, 'voleibol/futevolei', inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(15, 'tênis', inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(16, 'dança (balé, dança de salão, dança do ventre)',
inplace = True)
df_atividadeSelecionadas['tipo_exercicio'].replace(17, 'outros', inplace = True)
df_atividadeSelecionadas['grau_escolaridade'].replace(1, 'curso primário', inplace = True)
df_atividadeSelecionadas['grau_escolaridade'].replace(2, 'admissão', inplace = True)
df_atividadeSelecionadas['grau_escolaridade'].replace(3, 'curso ginásio ou ginásio', inplace = True)
df_atividadeSelecionadas['grau_escolaridade'].replace(4, '1º grau ou fundamental ou supletivo de 1º
grau', inplace = True)
df_atividadeSelecionadas['grau_escolaridade'].replace(5, '2º grau ou colégio ou técnico ou normal ou
científico ou ensino médio ou supletivo de 2º grau', inplace = True)
df_atividadeSelecionadas['grau_escolaridade'].replace(6, '3º grau ou curso superior', inplace = True)
df_atividadeSelecionadas['grau_escolaridade'].replace(7, 'pós-graduação (especialização, mestrado,
doutorado)', inplace = True)
df_atividadeSelecionadas['grau_escolaridade'].replace(8, 'nunca estudou', inplace = True)
df_atividadeSelecionadas['grau_escolaridade'].replace(777, 'não sabe', inplace = True)
df_atividadeSelecionadas['grau_escolaridade'].replace(888, 'não quis responder', inplace = True)
df_atividadeSelecionadas['civil'].replace(1, 'solteiro', inplace = True)
df_atividadeSelecionadas['civil'].replace(2, 'casado legalmente', inplace = True)
df_atividadeSelecionadas['civil'].replace(3, 'tem união estável há mais de seis meses', inplace = True)
df_atividadeSelecionadas['civil'].replace(4, 'viúvo', inplace = True)
df_atividadeSelecionadas['civil'].replace(5, 'separado ou divorciado', inplace = True)
df_atividadeSelecionadas['civil'].replace(888, 'não quis informar', inplace = True)
#formate a coluna frequencia exercicio

```

```

df_atividadeSelecionadas['frequencia_exercicio'].replace(1,'1 a 2 dias por semana', inplace =True)
df_atividadeSelecionadas['frequencia_exercicio'].replace(2,'3 a 4 dias por semana', inplace =True)
df_atividadeSelecionadas['frequencia_exercicio'].replace(3,'5 a 6 dias por semana', inplace =True)
df_atividadeSelecionadas['frequencia_exercicio'].replace(4,'todos os dias ( inclusive sábado e domingo)',
inplace =True)
#formate a coluna duracao exercicio
df_atividadeSelecionadas['duracao_exercicio'].replace(1,'menos que 10 minutos', inplace =True)
df_atividadeSelecionadas['duracao_exercicio'].replace(2,'entre 10 e 19 minutos', inplace =True)
df_atividadeSelecionadas['duracao_exercicio'].replace(3,'entre 20 e 29 minutos', inplace =True)
df_atividadeSelecionadas['duracao_exercicio'].replace(4,'entre 30 e 39 minutos', inplace =True)
df_atividadeSelecionadas['duracao_exercicio'].replace(5,'entre 40 e 49 minutos', inplace =True)
df_atividadeSelecionadas['duracao_exercicio'].replace(6,'entre 50 e 59 minutos', inplace =True)
df_atividadeSelecionadas['duracao_exercicio'].replace(7,'60 minutos ou mais', inplace =True)
#formate a coluna cor
df_atividadeSelecionadas['cor'].replace(1,'branca', inplace =True)
df_atividadeSelecionadas['cor'].replace(2,'preta', inplace =True)
df_atividadeSelecionadas['cor'].replace(3,'amarela', inplace =True)
df_atividadeSelecionadas['cor'].replace(4,'parda', inplace =True)
df_atividadeSelecionadas['cor'].replace(5,'indígena', inplace =True)
df_atividadeSelecionadas['cor'].replace(80,'Morena', inplace =True)
df_atividadeSelecionadas['cor'].replace(777,'não sabe', inplace =True)
df_atividadeSelecionadas['cor'].replace(888,'não quis informar', inplace =True)

df_atividadeSelecionadas['sexo'].replace(1,'masculino', inplace =True)
df_atividadeSelecionadas['sexo'].replace(2,'feminino', inplace =True)

df = df_atividadeSelecionadas
#df_atividade_junt = df.drop(['chave'], axis = 1) # remove coluna
# Clean column names
df.columns = [clean_column(col) for col in df.columns]
return df.iloc[:1000000]

@test
def test_number_of_columns(df, *args) -> None:
    assert len(df.columns) >= 1, 'There needs to be at least 11 columns.'

#Gravação Dados Atividade Física
from mage_ai.settings.repo import get_repo_path
from mage_ai.io.config import ConfigFileLoader
from mage_ai.io.postgres import Postgres
from pandas import DataFrame
from os import path
if 'data_exporter' not in globals():
    from mage_ai.data_preparation.decorators import data_exporter
@data_exporter
def export_data_to_postgres(df: DataFrame, **kwargs) -> None:
    """
    Template for exporting data to a PostgreSQL database.
    Specify your configuration settings in 'io_config.yaml'.
    Docs: https://docs.mage.ai/design/data-loading#postgresql
    """

    schema_name = 'PUC_TCC_Atividade_Fisica' # Specify the name of the schema to export data to
    table_name = 'Atividade_Fisica' # Specify the name of the table to export data to
    config_path = path.join(get_repo_path(), 'io_config.yaml')

```

```

config_profile = 'default'
with Postgres.with_config(ConfigFileLoader(config_path, config_profile)) as loader:
    loader.export(
        df,
        schema_name,
        table_name,
        index=False, # Specifies whether to include index in exported table
        if_exists='replace', # Specify resolution policy if table name already exists
    )

#Carregamento Dados Cidade
import io
import pandas as pd
import requests
@data_loader
def load_data_from_api(*args, **kwargs):
    url = 'https://raw.githubusercontent.com/Gabriellr/TCC-Engenharia-de-Dados-PUC-
/main/Capitais_Censo.csv'
    response = requests.get(url)
    return pd.read_csv(io.StringIO(response.text), sep=',')
@test
def test_row_count(df, *args) -> None:
    assert len(df.index) >= 1, 'Verificar se os dados possuem linhas suficientes'

#Transformação Dados Cidade
def number_of_rows_per_key(df, key, column_name):
    data = df.groupby(key)[key].agg(['count'])
    data.columns = [column_name]
    return data
def clean_column(column_name):
    return column_name.lower().replace(' ', '_')
@transformer
def transform(df, *args, **kwargs):
#criar coluna com o calculo PIB dataframe df_censoSelecionadas
    df_censo = df
    #Criar lista das colunas que vão ser usadas Censo
    ColunasSelecionadaCenso = ['cidade', 'uf', 'nome_capital', 'populacao_estimada_pessoas',
'pib_per_capita', 'regiao', 'ano', 'Salario_medio_mensal']
    #Filtra para o dataframe novo somente as colunas que foram selecionadas
    df_censoSelecionadas = df_censo.filter(items=ColunasSelecionadaCenso )
    df = df_censoSelecionadas
    # Clean column names
    df.columns = [clean_column(col) for col in df.columns]
    return df.iloc[:1000000]
@test
def test_number_of_columns(df, *args) -> None:
    assert len(df.columns) >= 1, 'There needs to be at least 11 columns.'

#Gravação Dados Cidade
from mage_ai.settings.repo import get_repo_path
from mage_ai.io.config import ConfigFileLoader
from mage_ai.io.postgres import Postgres
from pandas import DataFrame
from os import path

```

```

if 'data_exporter' not in globals():
    from mage_ai.data_preparation.decorators import data_exporter
    @data_exporter
    def export_data_to_postgres(df: DataFrame, **kwargs) -> None:
        """
        Template for exporting data to a PostgreSQL database.
        Specify your configuration settings in 'io_config.yaml'.

        Docs: https://docs.mage.ai/design/data-loading#postgresql
        """
        schema_name = 'PUC_TCC_Atividade_Fisica' # Specify the name of the schema to export data to
        table_name = 'Cidade' # Specify the name of the table to export data to
        config_path = path.join(get_repo_path(), 'io_config.yaml')
        config_profile = 'default'

        with Postgres.with_config(ConfigFileLoader(config_path, config_profile)) as loader:
            loader.export(
                df,
                schema_name,
                table_name,
                index=False, # Specifies whether to include index in exported table
                if_exists='replace', # Specify resolution policy if table name already exists
            )

```

#Criação Tabelas Atividade Física e Cidade

```

CREATE TABLE IF NOT EXISTS puc_tcc_atividade_fisica.atividade_fisica
(
    ano smallint,
    cidade smallint,
    civil text COLLATE pg_catalog."default",
    idade smallint,
    sexo text COLLATE pg_catalog."default",
    grau_escolaridade text COLLATE pg_catalog."default",
    peso smallint,
    altura smallint,
    pratica_exercicio text COLLATE pg_catalog."default",
    tipo_exercicio text COLLATE pg_catalog."default",
    pratica_exercicio_1_vez_na_semana double precision,
    frequencia_exercicio text COLLATE pg_catalog."default",
    duracao_exercicio text COLLATE pg_catalog."default",
    cor text COLLATE pg_catalog."default",
    imc double precision
)

```

```
CREATE TABLE IF NOT EXISTS puc_tcc_atividade_fisica.cidade
(
  cidade smallint,
  uf text COLLATE pg_catalog."default",
  nome_capital text COLLATE pg_catalog."default",
  pib_per_capita text COLLATE pg_catalog."default",
  regioao text COLLATE pg_catalog."default",
  ano smallint,
  salario_medio_mensal text COLLATE pg_catalog."default"
)
```