**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS NÚCLEO DE EDUCAÇÃO A DISTÂNCIA**

**Pós-graduação *Lato Sensu* em Engenharia de Dados**

**Leonardo Nascimento de Souza Sepulveda**

**ANALISE PRÁTICA DE REGISTOS DE BOLETINS DE OCORRENCIA**

**NO ESTADO DO RIO DE JANEIRO**

São Lourenço – MG 2024

**Leonardo Nacimento de Souza Sepulveda**

**ANALISE PRÁTICA DE REGISTOS DE BOLETINS DE OCORRENCIA**

**NO ESTADO DO RIO DE JANEIRO**

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

São Lourenço -

MG 2024

**SUMÁRIO**

1. [Introdução 4](#_bookmark0)
   1. [Contextualização 4](#_bookmark1)
   2. [O problema proposto](#_bookmark2) 5
2. [Coleta de Dados](#_bookmark3) 6
3. [Ingestão/Processamento/Tratamento de Dados](#_bookmark4) 9
4. [Orquestração de Dados](#_bookmark5) 13
5. [Visualização de dados](#_bookmark6) 18
6. [Apresentação dos Resultados](#_bookmark7) 25
7. [Links](#_bookmark8) 26
8. Referências 26

[APÊNDICE](#_bookmark9) 27

## Introdução

## Contextualização

A polícia acompanha de perto as taxas de delitos por várias razões cruciais que afetam diretamente sua capacidade de proteger e servir a comunidade. Aqui estão alguns pontos essenciais que destacam a importância dessa prática:

Identificação de Tendências Criminais: Monitorar as taxas de delitos ao longo do tempo ajuda a identificar tendências criminais em uma determinada área geográfica. Essas tendências podem incluir aumentos repentinos em certos tipos de crimes ou mudanças nos padrões de criminalidade sazonal.

Alocação de Recursos: Ao compreender as taxas de delitos em diferentes regiões, a polícia pode alocar recursos de forma mais eficaz. Por exemplo, áreas com altas taxas de criminalidade podem exigir mais patrulhamento e presença policial para prevenir crimes e manter a ordem pública.

Planejamento Estratégico: As informações sobre as taxas de delitos são fundamentais para o desenvolvimento de estratégias policiais eficazes. Isso pode incluir a implementação de programas de prevenção ao crime, o foco em áreas de alto risco e a colaboração com outras agências para abordar questões específicas de segurança.

Avaliação de Políticas Públicas: O acompanhamento das taxas de delitos permite avaliar a eficácia das políticas públicas relacionadas à segurança. Com base nos dados coletados, as autoridades podem ajustar suas abordagens e implementar novas políticas para lidar com desafios emergentes.

Resposta a Emergências: O conhecimento das taxas de delitos é vital para a capacidade de resposta da polícia a emergências e crises. Isso inclui a mobilização de recursos adicionais em resposta a eventos como surtos de violência, desastres naturais ou situações de segurança pública.

Promoção da Segurança Pública: Por fim, o acompanhamento das taxas de delitos desempenha um papel fundamental na promoção da segurança pública e no fortalecimento da confiança da comunidade na polícia. Quando os cidadãos percebem que a polícia está ativamente envolvida na prevenção e na redução da criminalidade, isso pode levar a uma maior cooperação e apoio da comunidade.

Em resumo, a polícia acompanha de perto as taxas de delitos para entender melhor as dinâmicas criminais, tomar decisões informadas, prevenir crimes e proteger a comunidade de maneira mais eficaz.

## O problema proposto

Neste trabalho, será utilizado a Ingestão Transformação e Gravação de dados para uma Análise Exploratória, para extração de informações importantes do acompanhamento da mancha criminal dos últimos anos

Assim, estará disponível uma base de dados para analisar os resultados e utilizá-los em decisões futuras.

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

* + - Realizar uma análise nos dados das da mancha criminal do Estado do Rio de Janeio. Desta forma, auxiliaremos para ver qual o comportamento do crime por cidades dentro do estado e seus respectivos delitos.
    - Os dados que serão analisados, foram coletados do (ISP - RJ) e IBGE.
    - As análises realizadas, têm como objetivo encontrar padrões, métricas e tendências que auxiliarão no entendimento das bases trabalhadas. E assim, poderemos indicar quais características dos crimes de uma cidade ou do estado como um todo.

¹ <https://basedosdados.org/dataset/90324ba8-9c39-4191-a8a4-302f93732464?table=edf80eb4-0c97-4d07-bf4a-711144510b81>

² <https://www.ibge.gov.br/geociencias/organizacao-do-territorio/estrutura-territorial/23701-divisao-territorial-brasileira.html>

## Coleta de Dados

O Instituo de Segurança Pública do Rio de Janeiro (ISP) fornece bases de dados de registros criminais e de atividade policial. As estatísticas divulgadas são construídas a partir dos Registros de Ocorrência (RO) lavrados nas delegacias da Secretaria de Estado de Polícia Civil do Rio de Janeiro (SEPOL), além de informações complementares de órgãos específicos da Secretaria de Estado de Polícia Militar do Rio de Janeiro (SEPM). Antes de serem consolidados no ISP, os RO são submetidos ao controle de qualidade realizado pela Corregedoria Geral de Polícia (CGPOL) da Secretaria de Estado de Polícia Civil. As estatísticas produzidas baseiam-se na data em que foi confeccionado o Registro de Ocorrência.

Vamos trabalhar aqui com 2 fontes de dados, a primeira do ISP que é a Taxa de Evolução Mensal dos Municípios e do IBGE o Relatório DTB Brasil Municipios.

Os campos de cada fonte de dados esta descrito abaixo conforme dicionário de dados:

Taxa de Evolução Mensal dos Municípios

Essa fonte de dados esta disponível em <https://basedosdados.org/dataset/90324ba8-9c39-4191-a8a4-302f93732464?table=d109b89d-7dd4-460f-b236-83eb7ade4fe9>

Os dados aqui abordados, estão disponibilizados no range de tempo de 2014 à 2023 e a métrica comparativa é a ***taxas por 100 mil habitantes***.

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descrição** |
| ano | INT64 | Ano da ocorrência |
| id\_municipio | STRING | Código IBGE de 7 dígitos do município |
| mes | INT64 | Mês da ocorrência |
| regiao | STRING | Região do Rio de Janeiro |
| taxa\_aaapai | INT64 | Número de autos de Apreensão de Adolescente por Prática de Ato Infracional (AAAPAI) por 100 mil habitantes |
| taxa\_ameaca | INT64 | Ameaça por 100 mil habitantes |
| taxa\_apf | INT64 | Número de autos de Prisão em Flagrante (APF) por 100 mil habitantes |
| taxa\_apreensao\_drogas | INT64 | Apreensão de drogas por 100 mil habitantes |
| taxa\_cmba | INT64 | Cumprimento de Mandado de Busca e Apreensão (CMB) por 100 mil habitantes |
| taxa\_cmp | INT64 | Cumprimento de Mandado de Prisão (CMP) por 100 mil habitantes |
| taxa\_crimes\_violentos\_letais\_intencionais | INT64 | Crimes Violentos Letais Intencionais (CVLI) por 100 mil habitantes |
| taxa\_encontro\_cadaver | INT64 | Encontro de cadáver por 100 mil habitantes |
| taxa\_encontro\_ossada | INT64 | Encontro de ossada por 100 mil habitantes |
| taxa\_estelionato | INT64 | Estelionato por 100 mil habitantes |
| taxa\_estupro | INT64 | Estupro por 100 mil habitantes |
| taxa\_extorsao | INT64 | Extorsão por 100 mil habitantes |
| taxa\_furto\_bicicleta | INT64 | Furto de bicicleta por 100 mil habitantes |
| taxa\_furto\_celular | INT64 | Furto de telefone celular por 100 mil habitantes |
| taxa\_furto\_coletivo | INT64 | Furto em coletivo por 100 mil habitantes |
| taxa\_furto\_transeunte | INT64 | Furto a transeunte por 100 mil habitantes |
| taxa\_furto\_veiculos | INT64 | Furto de veículo por 100 mil habitantes |
| taxa\_homicidio\_culposo | INT64 | Homicídio culposo (trânsito) por 100 mil habitantes |
| taxa\_homicidio\_doloso | INT64 | Homicídio doloso por 100 mil habitantes |
| taxa\_homicidio\_intervencao\_policial | INT64 | Morte por intervenção de agente do Estadopor 100 mil habitantes |
| taxa\_latrocinio | INT64 | Latrocínio por 100 mil habitantes |
| taxa\_lesao\_corporal\_culposa | INT64 | Lesão corporal culposa (trânsito) por 100 mil habitantes |
| taxa\_lesao\_corporal\_dolosa | INT64 | Lesão corporal dolosa por 100 mil habitantes |
| taxa\_lesao\_corporal\_morte | INT64 | Lesão corporal seguida de morte por 100 mil habitantes |
| taxa\_letalidade\_violenta | INT64 | Letalidade violenta por 100 mil habitantes |
| taxa\_outros\_furtos | INT64 | Outros furtos que não os listados acima por 100 mil habitantes |
| taxa\_outros\_roubos | INT64 | Outros roubos que não os listados acima por 100 mil habitantes |
| taxa\_pessoas\_desaparecidas | INT64 | Pessoas desaparecidas por 100 mil habitantes |
| taxa\_policial\_civil\_morto\_servico | INT64 | Policiais Civis mortos em serviço por 100 mil habitantes |
| taxa\_policial\_militar\_morto\_servico | INT64 | Policiais Militares mortos em serviço por 100 mil habitantes |
| taxa\_registro\_apreensao\_drogas\_sem\_autor | INT64 | Número de registros que possuem algum título referente a apreensão de drogas sem autor por 100 mil habitantes |
| taxa\_registro\_ocorrencia | INT64 | Registro de ocorrências por 100 mil habitantes |
| taxa\_registro\_posse\_drogas | INT64 | Número de registros que possuem algum título referente a posse de drogas por 100 mil habitantes |
| taxa\_registro\_trafico\_drogas | INT64 | Número de registros que possuem algum título referente a tráfico de drogas por 100 mil habitantes |
| taxa\_registro\_veiculo\_recuperado | INT64 | Recuperação de veículo por 100 mil habitantes |
| taxa\_roubo\_apos\_saque | INT64 | Roubo após saque em instituição financeira por 100 mil habitantes |
| taxa\_roubo\_banco | INT64 | Roubo a banco por 100 mil habitantes |
| taxa\_roubo\_bicicleta | INT64 | Roubo de bicicleta por 100 mil habitantes |
| taxa\_roubo\_caixa\_eletronico | INT64 | Roubo de caixa eletrônico por 100 mil habitantes |
| taxa\_roubo\_carga | INT64 | Roubo de carga por 100 mil habitantes |
| taxa\_roubo\_celular | INT64 | Roubo ao telefone celular por 100 mil habitantes |
| taxa\_roubo\_comercio | INT64 | Roubo a estabelecimento comercial por 100 mil habitantes |
| taxa\_roubo\_conducao\_saque | INT64 | Roubo com condução da vítima para saque em instituição financeira por 100 mil habitantes |
| taxa\_roubo\_corporal\_coletivo | INT64 | Roubo em coletivo por 100 mil habitantes |
| taxa\_roubo\_residencia | INT64 | Roubo a residência por 100 mil habitantes |
| taxa\_roubo\_rua | INT64 | Roubo de rua por 100 mil habitantes |
| taxa\_roubo\_transeunte | INT64 | Roubo a transeunte por 100 mil habitantes |
| taxa\_roubo\_veiculo | INT64 | Roubo de veículo por 100 mil habitantes |
| taxa\_sequestro | INT64 | Extorsão mediante sequestro (sequestro clássico) por 100 mil habitantes |
| taxa\_sequestro\_relampago | INT64 | Extorsão com momentânea privação da liberdade (sequestro relâmpago) por 100 mil habitantes |
| taxa\_tentativa\_homicidio | INT64 | Tentativa de homicídio por 100 mil habitantes |
| taxa\_total\_furtos | INT64 | Total de furtos por 100 mil habitantes |
| taxa\_total\_roubos | INT64 | Total de roubos por 100 mil habitantes |
| tipo\_fase | STRING | Tipo de fase (e.g. Consolidado sem errata, Consolidado com errata) |

Nível bservação

|  |  |
| --- | --- |
| **Entidade** | **Colunas Correspontedentes** |
| Ano | ano |
| Município | Id\_ municipio |
| Mês | mes |

Relatório DTB Brasil Municipios

Essa fonte de dados esta disponivel em <https://www.ibge.gov.br/geociencias/organizacao-do-territorio/estrutura-territorial/23701-divisao-territorial-brasileira.html>, pós o download da base, dentro do arquivo ZIP, teremos o relatório em questão.

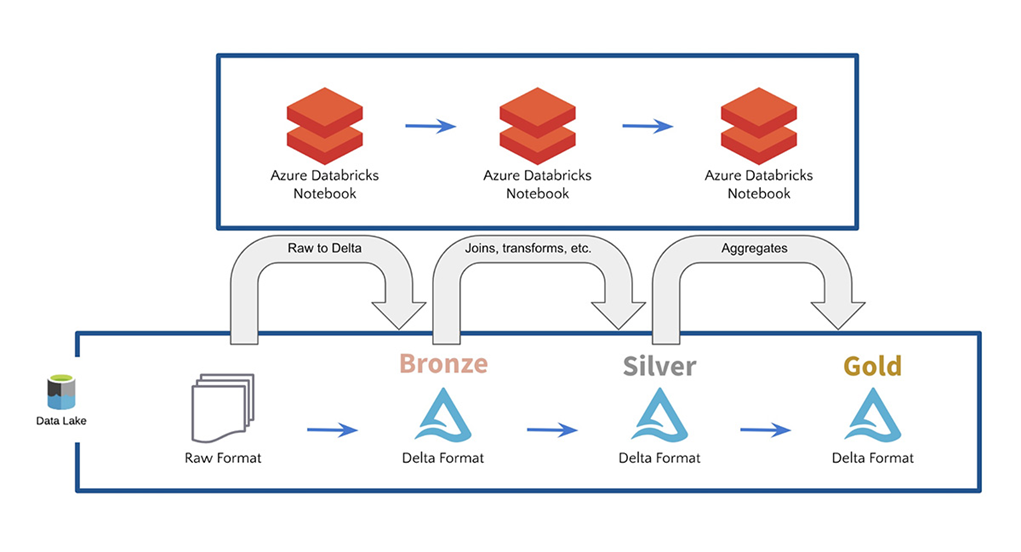
|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descrição** |
| UF | STRING | Código da Unidade Federativa |
| Nome\_UF | STRING | Nome da Unidade Federativa |
| Regiao\_Geografica\_Intermediaria | STRING | Código Regiao Geografica Intermediaria |
| Nome\_Regiao\_Geografica\_Intermediaria | STRING | Nome Regiao Geografica Intermediaria |
| Regiao\_Geografica\_Imediata | STRING | Código Regiao Geografica Imediata |
| Nome\_Regiao\_Geografica\_Imediata | STRING | Nome Regiao Geografica Imediata |
| MesoRegiao\_Geografica | STRING | Código MesoRegiao Geografica |
| Nome\_MesoRegiao | STRING | Nome MesoRegiao |
| MicroRegiao\_Geografica | STRING | Código MicroRegiao Geografica |
| Nome\_MicroRegiao | STRING | Nome MicroRegiao |
| Codigo\_Municipio | STRING | Codigo Municipio |
| Id\_Municipio | STRING | Código Municipio |
| Nome\_Municipio | STRING | Nome Municipio |

## Ingestão/Processamento/Tratamento de Dados

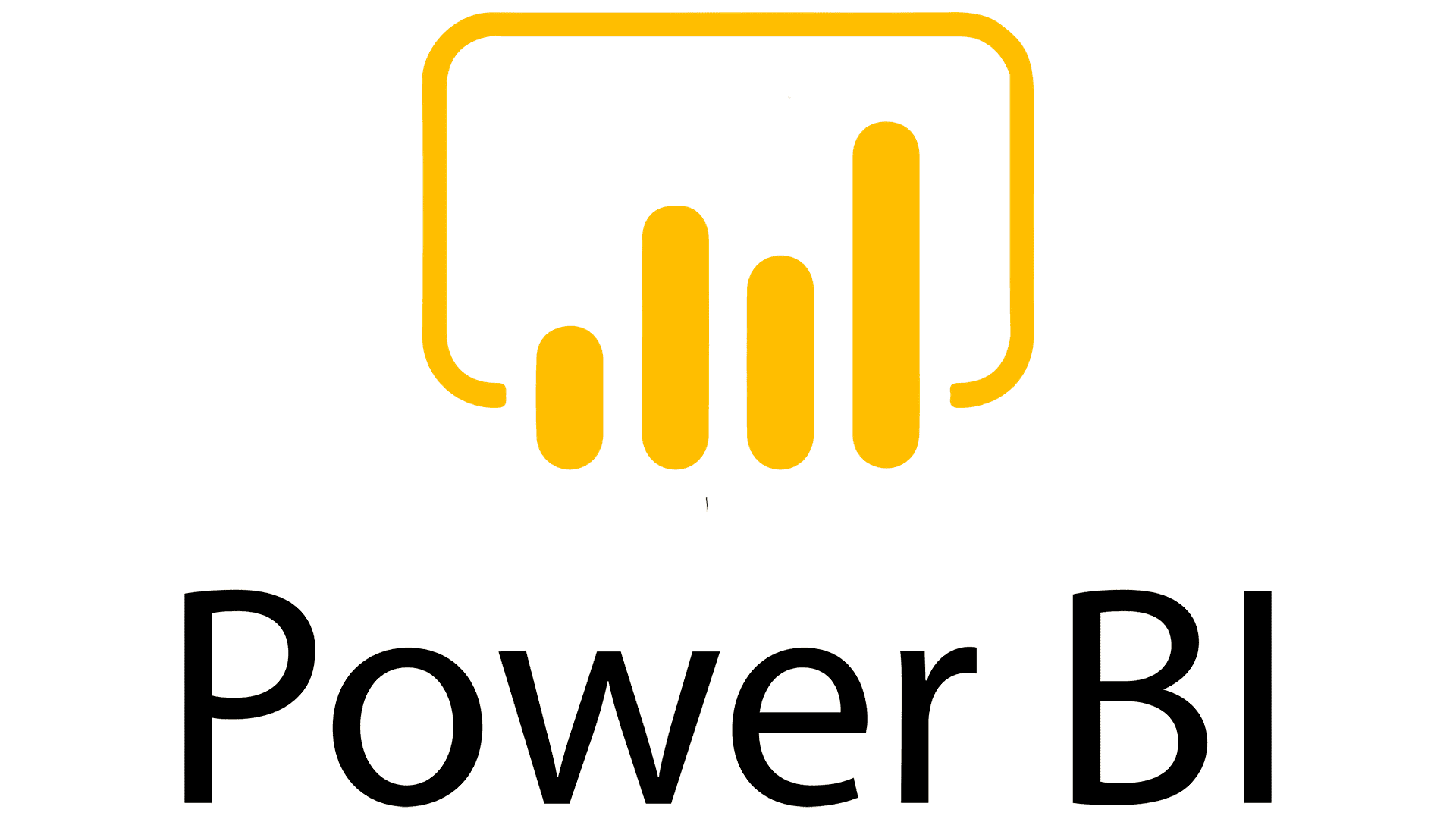
Nessa seção será apresentado o fluxo deingestão e processamento dos dados com as respctivas ferramentas utilizadas.

Todo o proecesso de ingestão, tranformação e armazenamento foi foi desenvolvido dentro do Microsoft Azure e foi utilizado o Microsoft Power BI para a camada Data Viz.

Na figura 1 temos a apresentação da Arquitetura aplicada

**Figura 1**: Screenshot do arquitetura do pipeline de dados



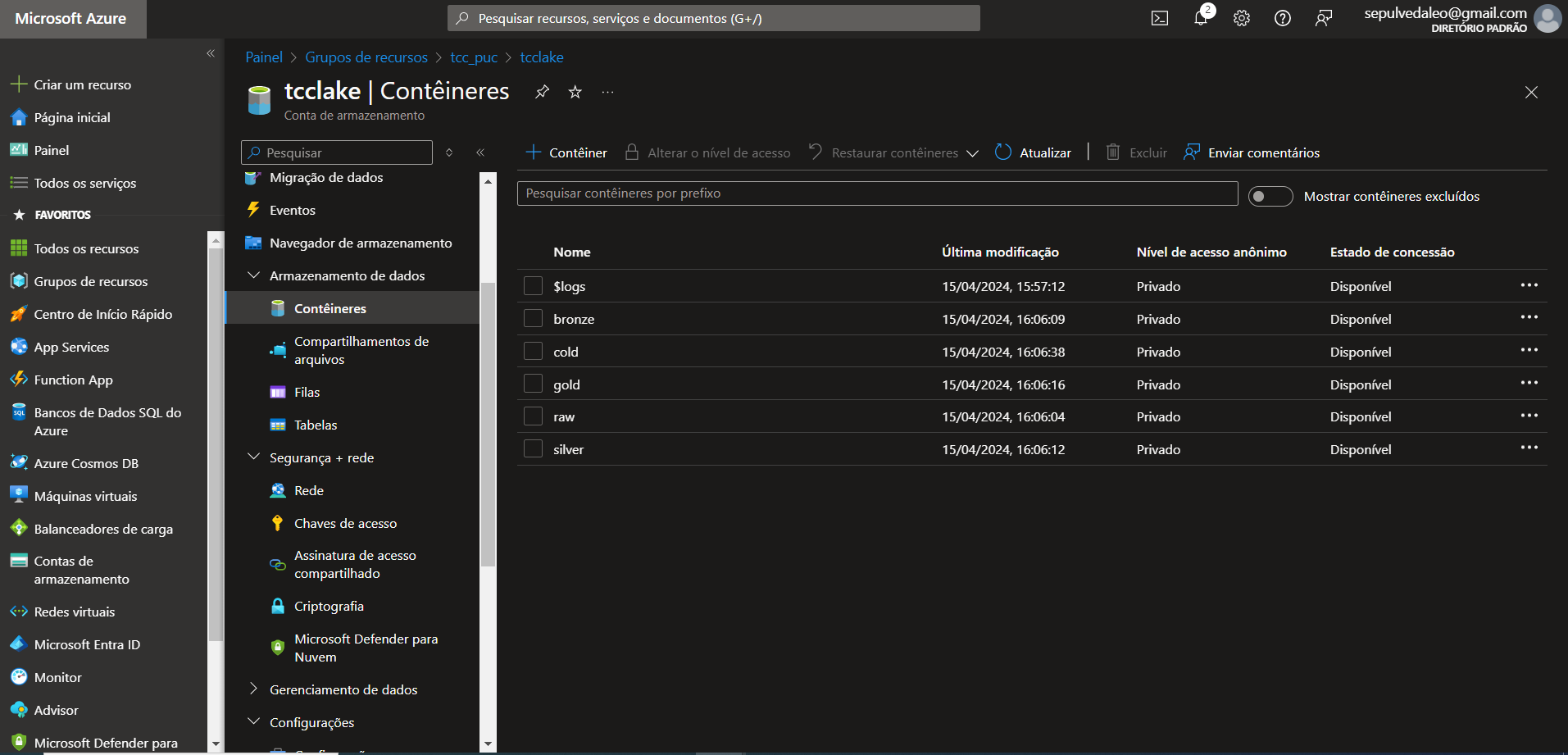


Fonte: <https://stackoverflow.com/questions/69956499/best-practice-for-layer-concept-with-adf-and-databricks>

e Autor

Foi escolhido a Plataforma Azure para o desenvolvimento do processo, nele utilizamos o Storage Account para armazenamento dos dados, o Databrcks para o processo de transformação e o Power BI para a camada Data Viz.

**Figura 2**: Screenshot do Azure Storage Account Estruturado



Fonte:Autor

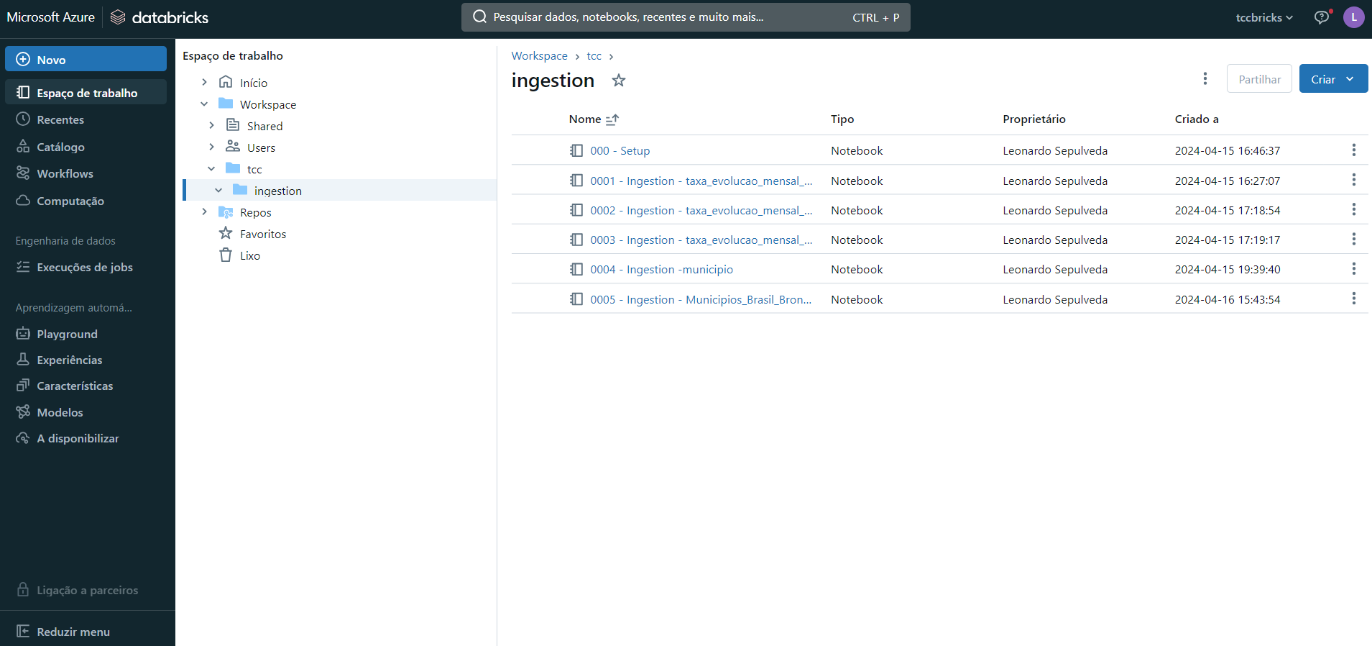
O Processo de ingestão e transformação dos dados, foi utilizado o Databricks com recursos de notebooks.

Foi criado dentro da Workspace DataBricks, uma pasta pra organização do projeto com o nome de tcc, dentro dessa pasta uma nova com o nome de ingestion com os respectivos notebooks.

Para a nomenclatura dos notebooks, foi utilizado uma boa prática de enumerar cada um deles para o fácil entendimento da sequencia de execução d processo até a camada Gold.

Foi respeitado a criação do notebook de de Setup, onde contém o script de create table, que servirá de apoio para um processo de CI/CD ou simples migração de ambiente, mantendo assim a integridade dos ambientes..

**Figura 3**: Screenshot da estruturação da Workspace DataBrciks.



Fonte: Autor

Dentro do processo de ingestão, os dados são recebidos o formato CSV, e em sua primeira etapa de tratamento, é convertido para formato DELTA (parquet) visando melhor performace e melhor intereração com a linguagem PySpark, que foi a linguagem utilizada no processo assim como a SQL.

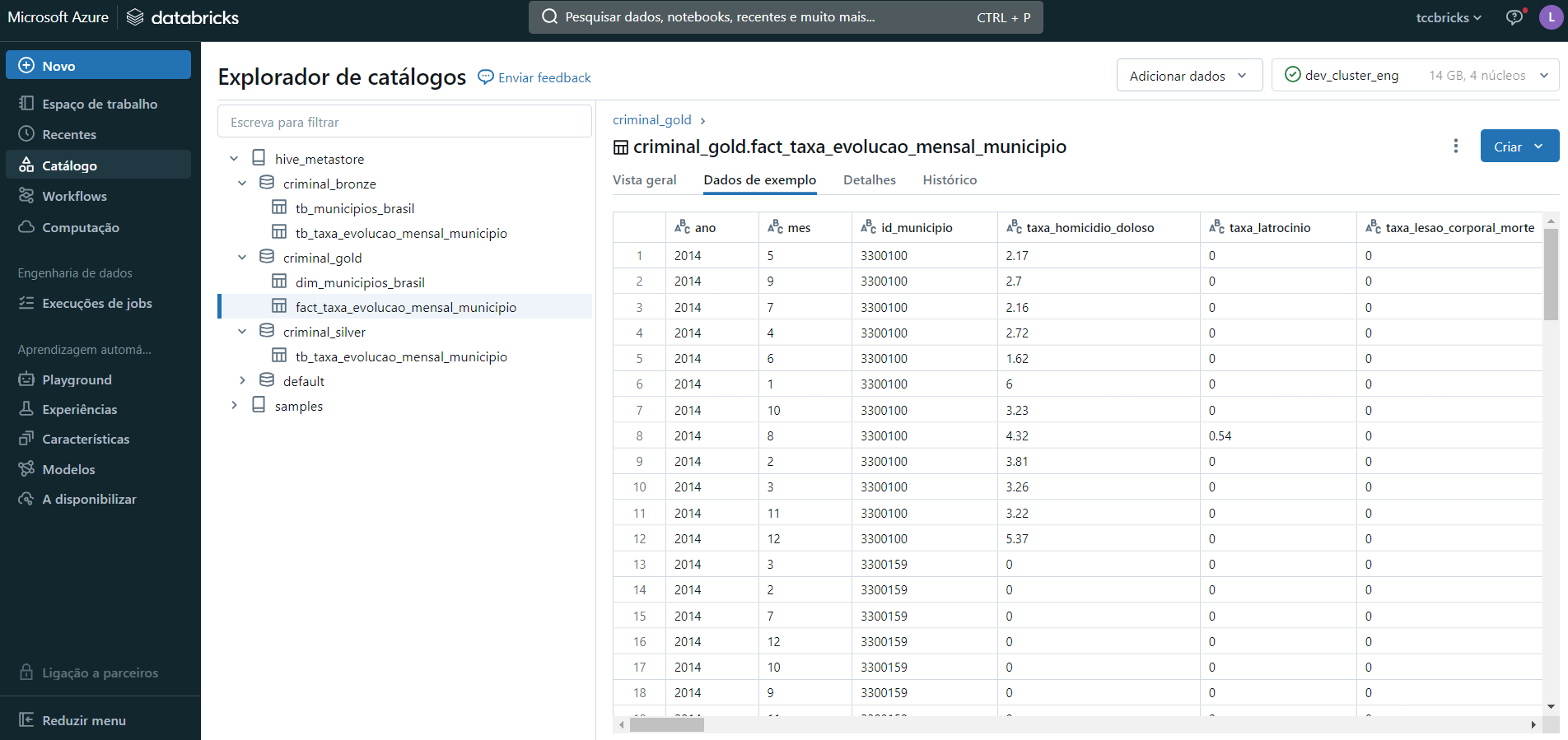
Além do armazenamento como DELTA table, foi disponibilizado também o acesso de forma simplificada, através do CATALOG, onde o usuário pode acessar os dados com a sensação de um banco convencional

**Figura 4**: Screenshot da estruturação doCatálogo DataBrciks.

Fonte: Autor

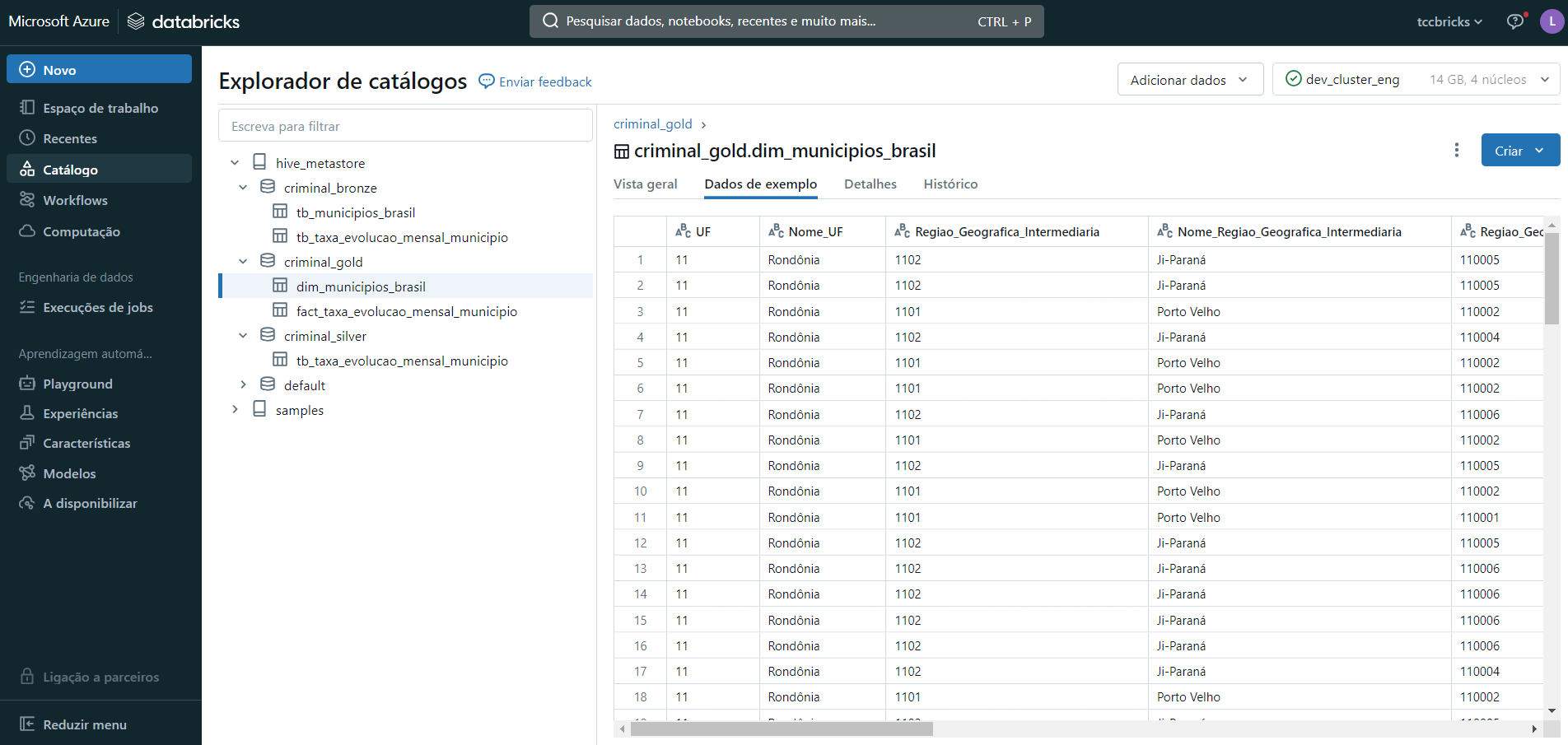
Esta disponibilizado um banco para cada camada do processo, ou seja, criminal\_bronze, silver e gold, agora é importante destacar, que na camada bronze e silver, a estrutura da tabela segue conceitos de modelagem, onde na etapa final, ou seja, na camada Gold, temos Fatos e dimensões.

**Figura 5**: Screenshot Script da tabela(criminal\_gold.fact\_taxa\_evolucao\_mensal\_municipio) criada no Gold.



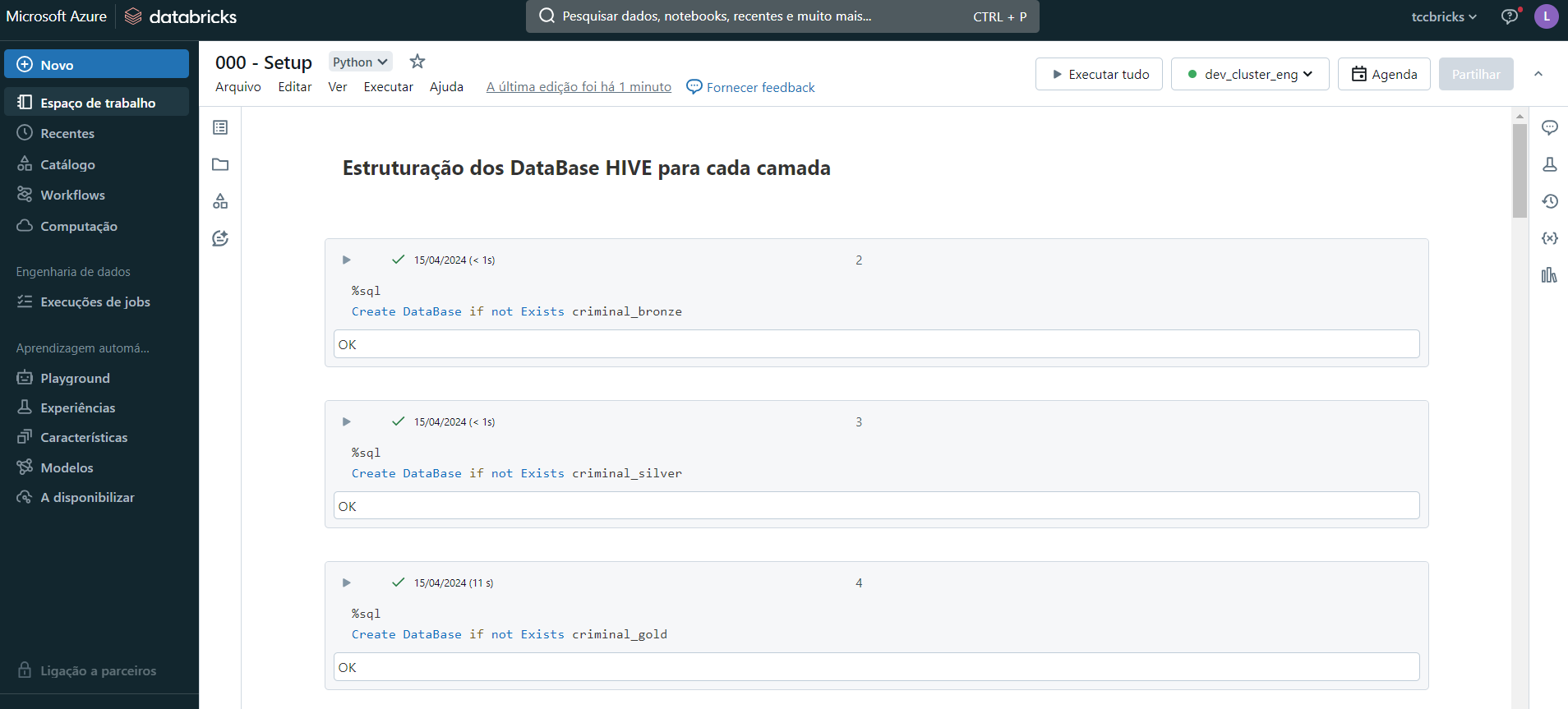
Fonte: Autor

**Figura 6**: Screenshot Script da (criminal\_gold.fact\_municipio\_brasil) criada no Gold.



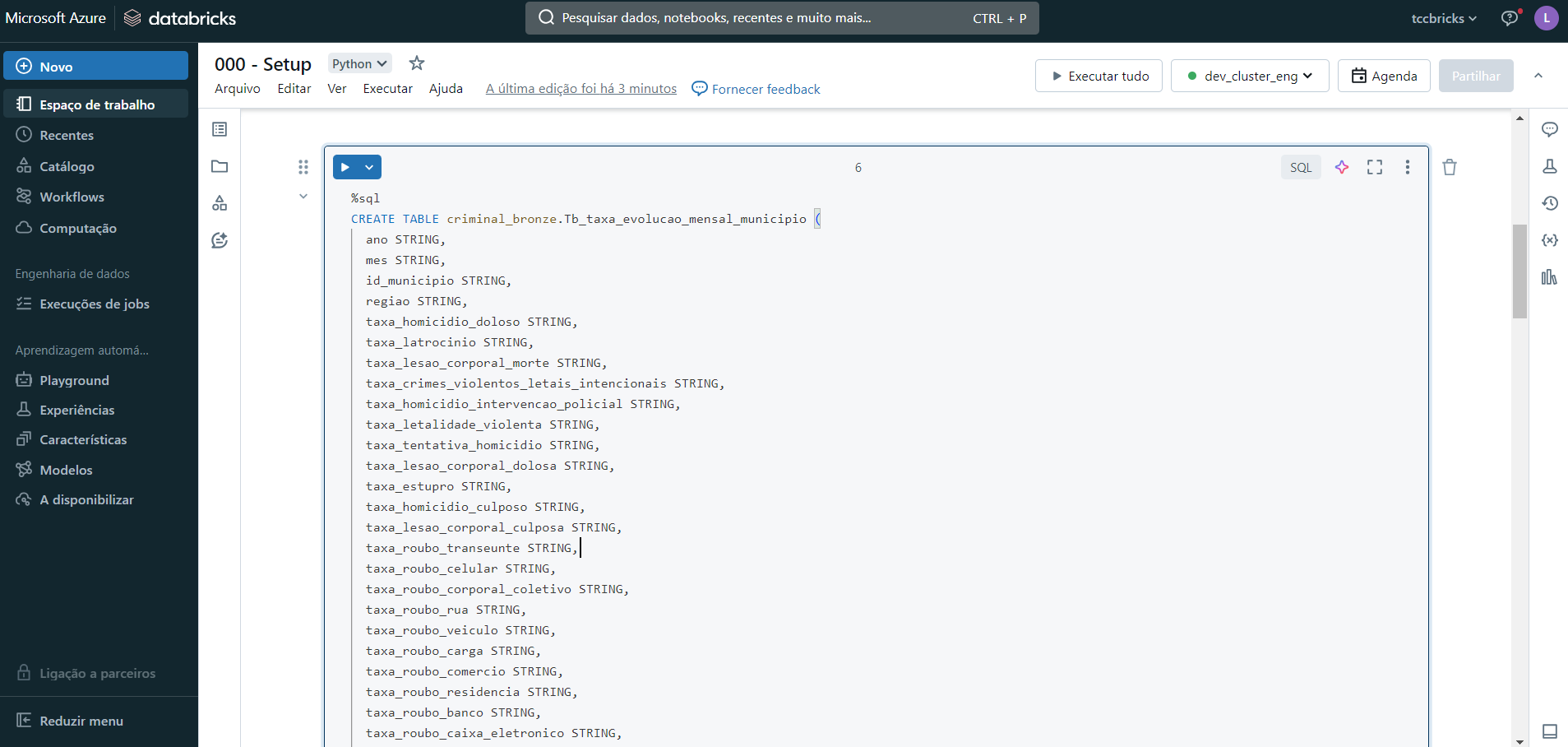
Fonte:Autor

**Figura 7**: Screenshot Script dos DataBase HIVE camadas Bonze, Silver e Gold PostgreSql.



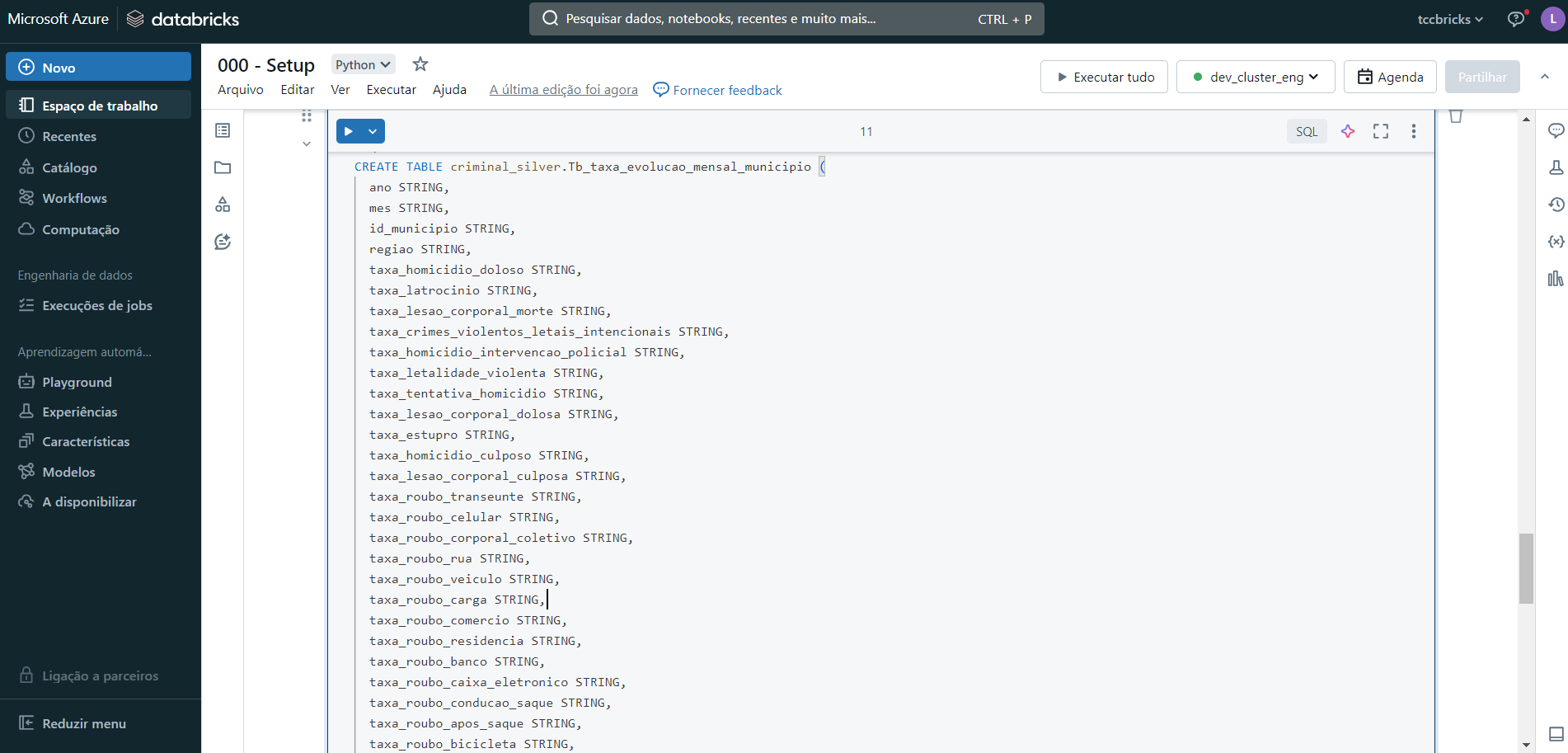
Fonte:Autor

**Figura 8**: Screenshot Script criação da tabela Tb\_taxa\_evolucao\_mensal\_municipio na camada Bonze



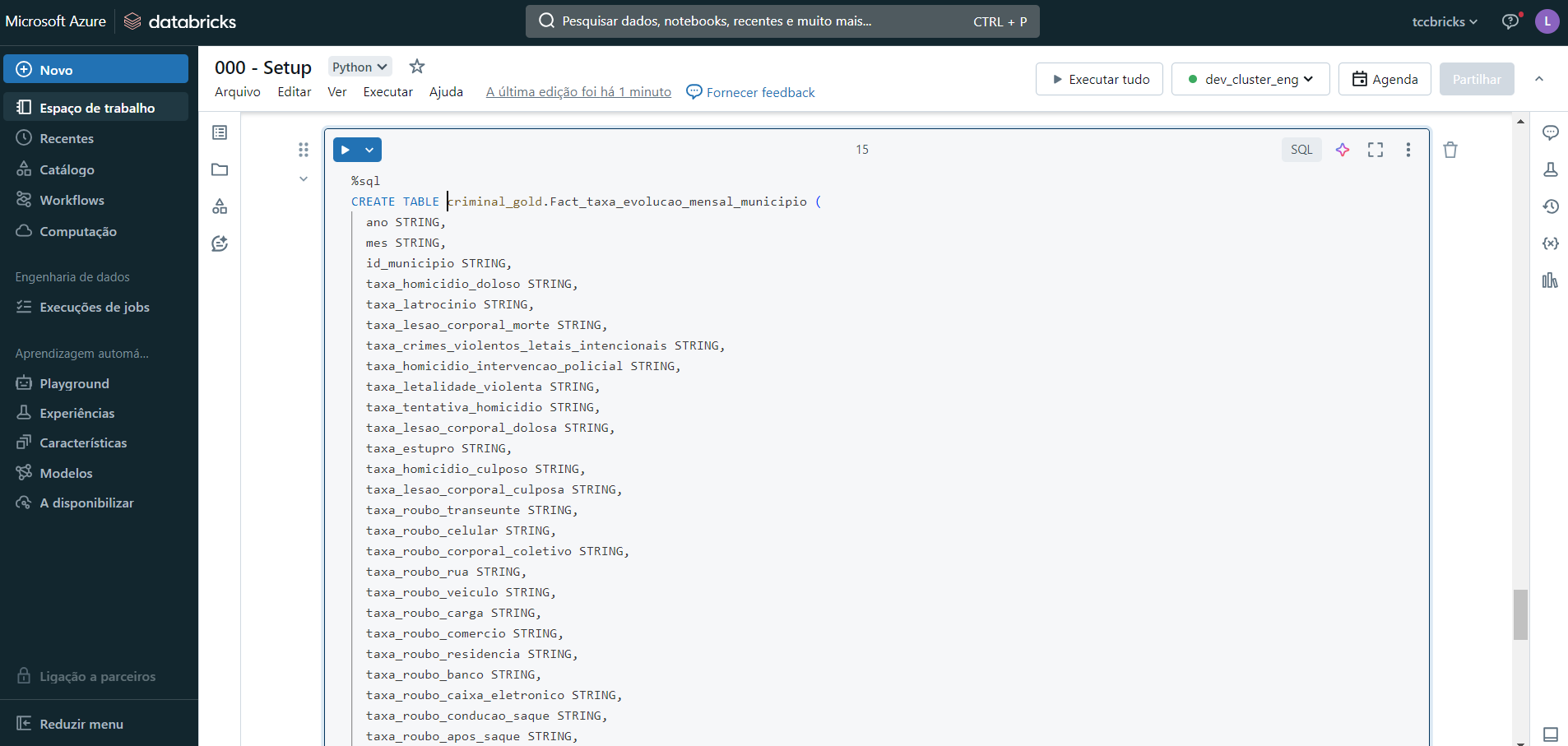
Fonte:Autor

**Figura 9**: Screenshot Script criação da tabela Tb\_taxa\_evolucao\_mensal\_municipio na camada Silver



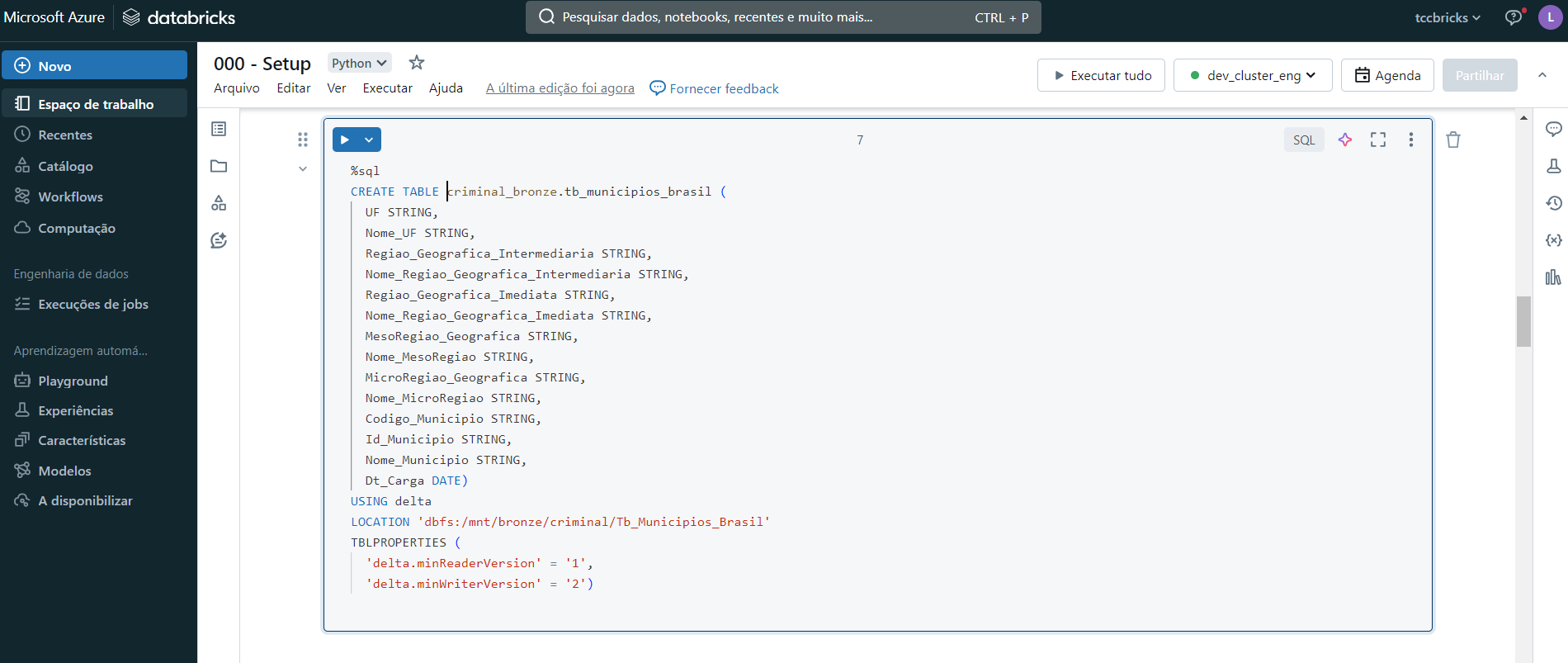
Fonte:Autor

**Figura 10**: Screenshot Script criação da tabela Tb\_taxa\_evolucao\_mensal\_municipio na camada Gold



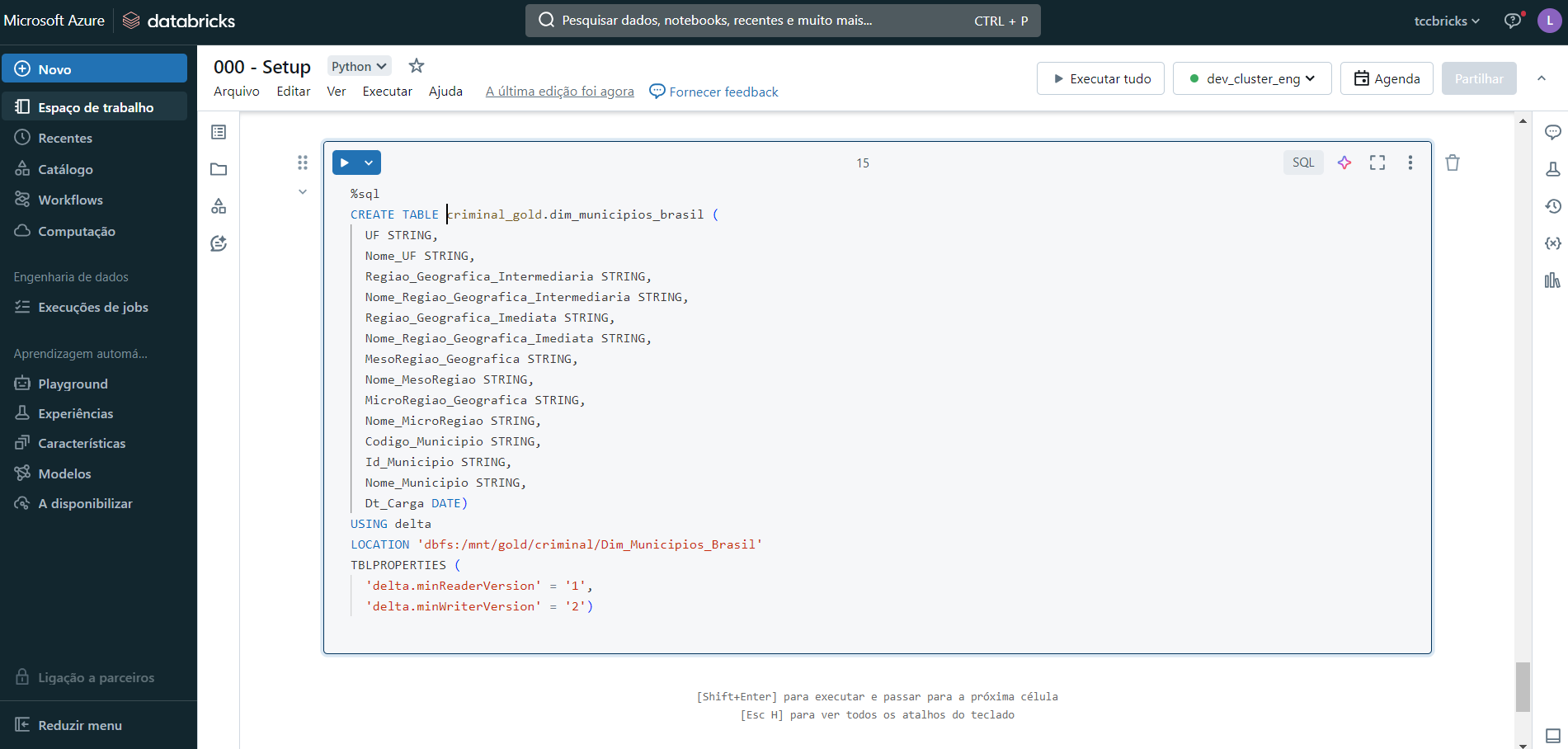
Fonte:Autor

**Figura 11**: Screenshot Script criação da tabela tb\_municipios\_brasil na camada Bronze



Fonte:Autor

**Figura 12**: Screenshot Script criação da tabela tb\_municipios\_brasil na camada Silver

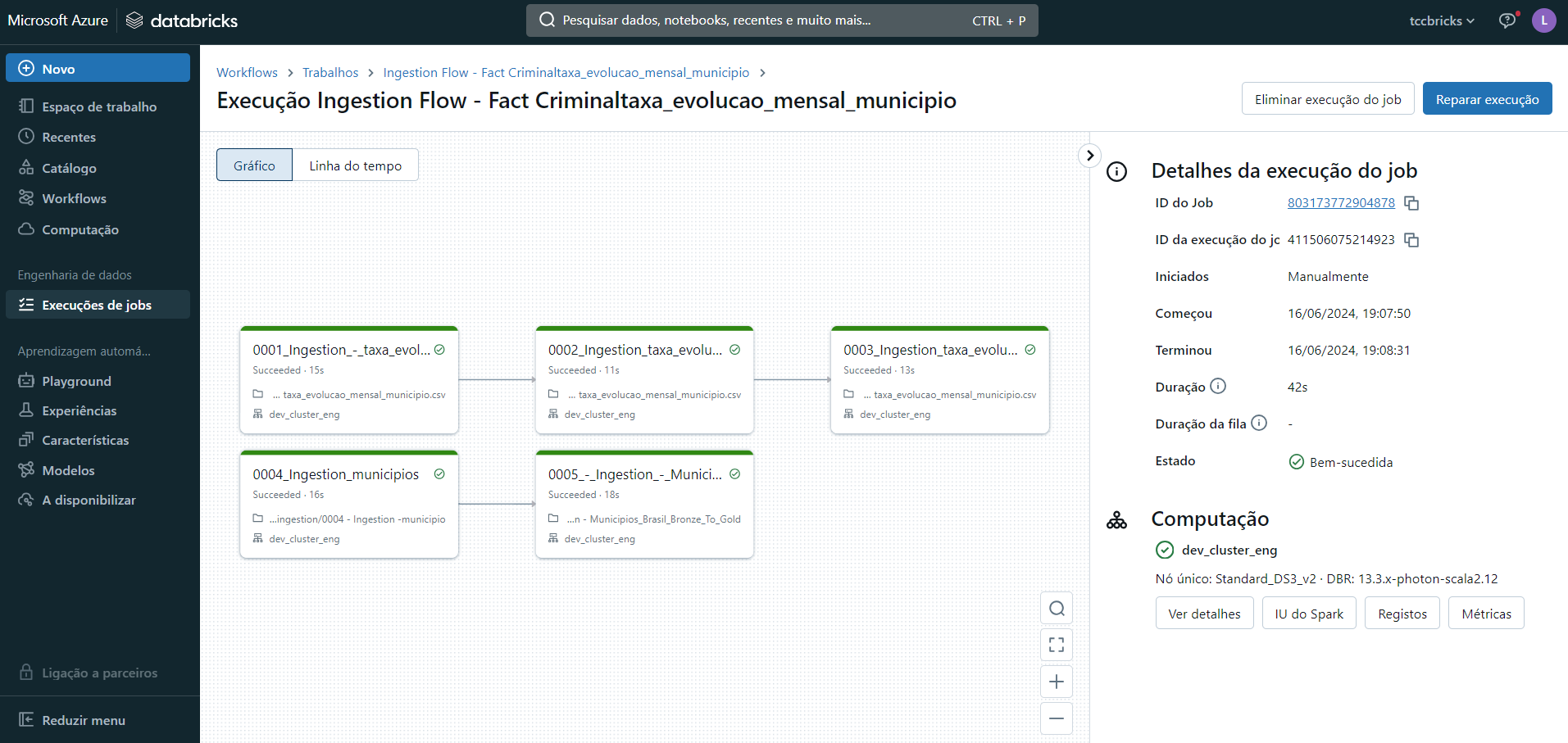


Fonte:Autor

Nota: Essa tabela, por se tratar de dados que sofrem alteração com baixissima regularidade, ela passa direto da camada bronze para a gold.

## Orquestração de Dados

Para a realização da orquestração de dados esta sendo utilizado o Workflow do DataBricks, ele passa a ser o responsavel pelo schedule de carga e garantir o fluxo do processo.

**Figura 13**: Screenshot Orquestração do projeto

Fonte:Autor

Notebooks do processo

**0001 - Ingestion - taxa\_evolucao\_mensal\_municipio.csv**

# Databricks notebook source

# MAGIC %md

# MAGIC ### Leitura do arquivo taxa\_evolucao\_mensal\_municipio.csv

# COMMAND ----------

csv = spark.read.format("csv").option("header", "true").option("sep", ",").option("encoding", "ISO-8859-1").load(f'/mnt/raw/taxa\_evolucao\_mensal\_municipio.csv').createOrReplaceTempView('taxa\_evolucao\_mensal\_municipio')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Variaves de controle de carga

# COMMAND ----------

from datetime import datetime, timedelta

current\_date = datetime.now()

max\_yearmonth\_date = current\_date

max\_yearmonth = max\_yearmonth\_date.strftime('%Y%m')

min\_yearmonth\_date = current\_date - timedelta(days=current\_date.day - 1) - timedelta(weeks=12)

min\_yearmonth = min\_yearmonth\_date.strftime('%Y%m')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Data preparation

# COMMAND ----------

query = spark.sql(f'''

Select

\*,

current\_Date() as dt\_carga

From taxa\_evolucao\_mensal\_municipio

Where 1=1

-- and Concat(ano, LPAD(mes, 2, '0')) >= {min\_yearmonth} Liberar quando houver mais dados na base

-- and Concat(ano, LPAD(mes, 2, '0')) <= {max\_yearmonth} Liberar quando houver mais dados na base

''')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Salvando os dados na camada bronze

# COMMAND ----------

path\_to\_write = '/mnt/bronze/criminal/Tb\_taxa\_evolucao\_mensal\_municipio'

query.write.mode("overwrite").option("mergeSchema", "true").option("path", path\_to\_write).saveAsTable(f"criminal\_bronze.Tb\_taxa\_evolucao\_mensal\_municipio")

**0002 - Ingestion - taxa\_evolucao\_mensal\_municipio\_Bronze\_To\_Silver**

# Databricks notebook source

# MAGIC %md

# MAGIC ### Variaves de controle de carga

# COMMAND ----------

from pyspark.sql import SparkSession

from datetime import datetime, timedelta

# COMMAND ----------

max\_yearmonth = spark.sql("""

Select Distinct Max(Concat(ano, LPAD(mes, 2, '0'))) as Year\_Month

From criminal\_bronze.Tb\_taxa\_evolucao\_mensal\_municipio

""").collect()[0][0]

min\_yearmonth = spark.sql('''Select Min(Concat(ano, LPAD(mes, 2, '0'))) as Year\_Month

From criminal\_bronze.Tb\_taxa\_evolucao\_mensal\_municipio

''').collect()[0][0]

# COMMAND ----------

# MAGIC %md

# MAGIC ### Limpeza do range de dado que será carregado

# COMMAND ----------

result = spark.sql(f"""Delete From criminal\_silver.Tb\_taxa\_evolucao\_mensal\_municipio

Where 1=1

and CONCAT(ano, LPAD(mes, 2, '0')) >= {min\_yearmonth}

and CONCAT(ano, LPAD(mes, 2, '0')) <= {max\_yearmonth}

"""

)

# COMMAND ----------

# MAGIC %md

# MAGIC ### Data preparation

# COMMAND ----------

query = spark.sql(f'''

Select

ano,

mes,

id\_municipio,

regiao,

taxa\_homicidio\_doloso,

taxa\_latrocinio,

taxa\_lesao\_corporal\_morte,

taxa\_crimes\_violentos\_letais\_intencionais,

taxa\_homicidio\_intervencao\_policial,

taxa\_letalidade\_violenta,

taxa\_tentativa\_homicidio,

taxa\_lesao\_corporal\_dolosa,

taxa\_estupro,

taxa\_homicidio\_culposo,

taxa\_lesao\_corporal\_culposa,

taxa\_roubo\_transeunte,

taxa\_roubo\_celular,

taxa\_roubo\_corporal\_coletivo,

taxa\_roubo\_rua,

taxa\_roubo\_veiculo,

taxa\_roubo\_carga,

taxa\_roubo\_comercio,

taxa\_roubo\_residencia,

taxa\_roubo\_banco,

taxa\_roubo\_caixa\_eletronico,

taxa\_roubo\_conducao\_saque,

taxa\_roubo\_apos\_saque,

taxa\_roubo\_bicicleta,

taxa\_outros\_roubos,

taxa\_total\_roubos,

taxa\_furto\_veiculos,

taxa\_furto\_transeunte,

taxa\_furto\_coletivo,

taxa\_furto\_celular,

taxa\_furto\_bicicleta,

taxa\_outros\_furtos,

taxa\_total\_furtos,

taxa\_sequestro,

taxa\_extorsao,

taxa\_sequestro\_relampago,

taxa\_estelionato,

taxa\_apreensao\_drogas,

taxa\_registro\_posse\_drogas,

taxa\_registro\_trafico\_drogas,

taxa\_registro\_apreensao\_drogas\_sem\_autor,

taxa\_registro\_veiculo\_recuperado,

taxa\_apf,

taxa\_aaapai,

taxa\_cmp,

taxa\_cmba,

taxa\_ameaca,

taxa\_pessoas\_desaparecidas,

taxa\_encontro\_cadaver,

taxa\_encontro\_ossada,

taxa\_policial\_militar\_morto\_servico,

taxa\_policial\_civil\_morto\_servico,

taxa\_registro\_ocorrencia,

tipo\_fase,

current\_Date() as dt\_carga

From criminal\_bronze.Tb\_taxa\_evolucao\_mensal\_municipio

Where 1=1

and CONCAT(ano, LPAD(mes, 2, '0')) >= '{min\_yearmonth}'

and CONCAT(ano, LPAD(mes, 2, '0')) <= '{max\_yearmonth}'

''')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Salvando os dados na camada bronze

# COMMAND ----------

path\_to\_write = '/mnt/silver/criminal/Tb\_taxa\_evolucao\_mensal\_municipio'

query.write.mode("append").option("mergeSchema", "true").option("path", path\_to\_write).saveAsTable(f"criminal\_silver.Tb\_taxa\_evolucao\_mensal\_municipio")

# COMMAND ----------

**0003 - Ingestion - taxa\_evolucao\_mensal\_municipio\_Silver\_To\_Gold**

# Databricks notebook source

from pyspark.sql import SparkSession

from datetime import datetime, timedelta

# COMMAND ----------

# MAGIC %md

# MAGIC ### Variaves de controle de carga

# COMMAND ----------

max\_yearmonth = spark.sql("""

Select Distinct Max(Concat(ano, LPAD(mes, 2, '0'))) as Year\_Month

From criminal\_bronze.Tb\_taxa\_evolucao\_mensal\_municipio

""").collect()[0][0]

min\_yearmonth = spark.sql('''Select Min(Concat(ano, LPAD(mes, 2, '0'))) as Year\_Month

From criminal\_bronze.Tb\_taxa\_evolucao\_mensal\_municipio

''').collect()[0][0]

# COMMAND ----------

# MAGIC %md

# MAGIC ### Limpeza do range de dado que será carregado

# COMMAND ----------

result = spark.sql(f"""Delete From criminal\_gold.Fact\_taxa\_evolucao\_mensal\_municipio

Where 1=1

and CONCAT(ano, LPAD(mes, 2, '0')) >= {min\_yearmonth}

and CONCAT(ano, LPAD(mes, 2, '0')) <= {max\_yearmonth}

"""

)

# COMMAND ----------

# MAGIC %md

# MAGIC ### Data preparation

# COMMAND ----------

query = spark.sql(f'''

Select

ano,

mes,

id\_municipio,

taxa\_homicidio\_doloso,

taxa\_latrocinio,

taxa\_lesao\_corporal\_morte,

taxa\_crimes\_violentos\_letais\_intencionais,

taxa\_homicidio\_intervencao\_policial,

taxa\_letalidade\_violenta,

taxa\_tentativa\_homicidio,

taxa\_lesao\_corporal\_dolosa,

taxa\_estupro,

taxa\_homicidio\_culposo,

taxa\_lesao\_corporal\_culposa,

taxa\_roubo\_transeunte,

taxa\_roubo\_celular,

taxa\_roubo\_corporal\_coletivo,

taxa\_roubo\_rua,

taxa\_roubo\_veiculo,

taxa\_roubo\_carga,

taxa\_roubo\_comercio,

taxa\_roubo\_residencia,

taxa\_roubo\_banco,

taxa\_roubo\_caixa\_eletronico,

taxa\_roubo\_conducao\_saque,

taxa\_roubo\_apos\_saque,

taxa\_roubo\_bicicleta,

taxa\_outros\_roubos,

taxa\_total\_roubos,

taxa\_furto\_veiculos,

taxa\_furto\_transeunte,

taxa\_furto\_coletivo,

taxa\_furto\_celular,

taxa\_furto\_bicicleta,

taxa\_outros\_furtos,

taxa\_total\_furtos,

taxa\_sequestro,

taxa\_extorsao,

taxa\_sequestro\_relampago,

taxa\_estelionato,

taxa\_apreensao\_drogas,

taxa\_registro\_posse\_drogas,

taxa\_registro\_trafico\_drogas,

taxa\_registro\_apreensao\_drogas\_sem\_autor,

taxa\_registro\_veiculo\_recuperado,

taxa\_apf,

taxa\_aaapai,

taxa\_cmp,

taxa\_cmba,

taxa\_ameaca,

taxa\_pessoas\_desaparecidas,

taxa\_encontro\_cadaver,

taxa\_encontro\_ossada,

taxa\_policial\_militar\_morto\_servico,

taxa\_policial\_civil\_morto\_servico,

taxa\_registro\_ocorrencia,

tipo\_fase,

current\_Date() as dt\_carga

From criminal\_silver.Tb\_taxa\_evolucao\_mensal\_municipio

Where 1=1

and CONCAT(ano, LPAD(mes, 2, '0')) >= '{min\_yearmonth}'

and CONCAT(ano, LPAD(mes, 2, '0')) <= '{max\_yearmonth}'

''')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Salvando os dados na camada bronze

# COMMAND ----------

path\_to\_write = '/mnt/gold/criminal/Fact\_taxa\_evolucao\_mensal\_municipio'

query.write.mode("overwrite").option("mergeSchema", "true").option("path", path\_to\_write).saveAsTable(f"criminal\_gold.Fact\_taxa\_evolucao\_mensal\_municipio")

# COMMAND ----------

**0004 - Ingestion -municipio**

# Databricks notebook source

# MAGIC %md

# MAGIC ### Leitura do arquivo taxa\_evolucao\_mensal\_municipio.csv

# COMMAND ----------

csv = spark.read.format("csv").option("header", "true").option("sep", ";").option("encoding", "UTF-8").load(f'/mnt/raw/RELATORIO\_DTB\_BRASIL\_MUNICIPIO.csv').createOrReplaceTempView('Municipios\_Brasil')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Data preparation

# COMMAND ----------

query = spark.sql(f'''

Select

UF as UF,

Nome\_UF as Nome\_UF,

`Região Geográfica Intermediária` as Regiao\_Geografica\_Intermediaria,

`Nome Região Geográfica Intermediária` as Nome\_Regiao\_Geografica\_Intermediaria,

`Região Geográfica Imediata` as Regiao\_Geografica\_Imediata,

`Nome Região Geográfica Imediata` as Nome\_Regiao\_Geografica\_Imediata,

`Mesorregião Geográfica` as MesoRegiao\_Geografica,

`Nome\_Mesorregião` as Nome\_MesoRegiao,

`Microrregião Geográfica` as MicroRegiao\_Geografica,

`Nome\_Microrregião` as Nome\_MicroRegiao,

`Município` as Codigo\_Municipio,

`Código Município Completo` as Id\_Municipio,

`Nome\_Município` as Nome\_Municipio,

current\_Date() as Dt\_Carga

From Municipios\_Brasil

Where 1=1

''')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Salvando os dados na camada bronze

# COMMAND ----------

path\_to\_write = '/mnt/bronze/criminal/Tb\_Municipios\_Brasil'

query.write.mode("overwrite").option("mergeSchema", "true").option("path", path\_to\_write).saveAsTable(f"criminal\_bronze.Tb\_Municipios\_Brasil")

# COMMAND ----------

# MAGIC %sql

# MAGIC Select \*

# MAGIC From Delta.`/mnt/bronze/criminal/Tb\_Municipios\_Brasil`

# COMMAND ----------

**0005 - Ingestion - Municipios\_Brasil\_Bronze\_To\_Gold**

# Databricks notebook source

# MAGIC %md

# MAGIC ### Leitura do arquivo taxa\_evolucao\_mensal\_municipio.csv

# COMMAND ----------

result = spark.sql(f"""Delete From criminal\_gold.Dim\_Municipios\_Brasil

Where 1=1

"""

)

# COMMAND ----------

# MAGIC %md

# MAGIC ### Data preparation

# COMMAND ----------

query = spark.sql(f'''

Select

UF,

Nome\_UF,

Regiao\_Geografica\_Intermediaria,

Nome\_Regiao\_Geografica\_Intermediaria,

Regiao\_Geografica\_Imediata,

Nome\_Regiao\_Geografica\_Imediata,

MesoRegiao\_Geografica,

Nome\_MesoRegiao,

MicroRegiao\_Geografica,

Nome\_MicroRegiao,

Codigo\_Municipio,

Id\_Municipio,

Nome\_Municipio,

Dt\_Carga

From criminal\_bronze.Tb\_Municipios\_Brasil

Where 1=1

''')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Salvando os dados na camada bronze

# COMMAND ----------

path\_to\_write = '/mnt/gold/criminal/Dim\_Municipios\_Brasil'

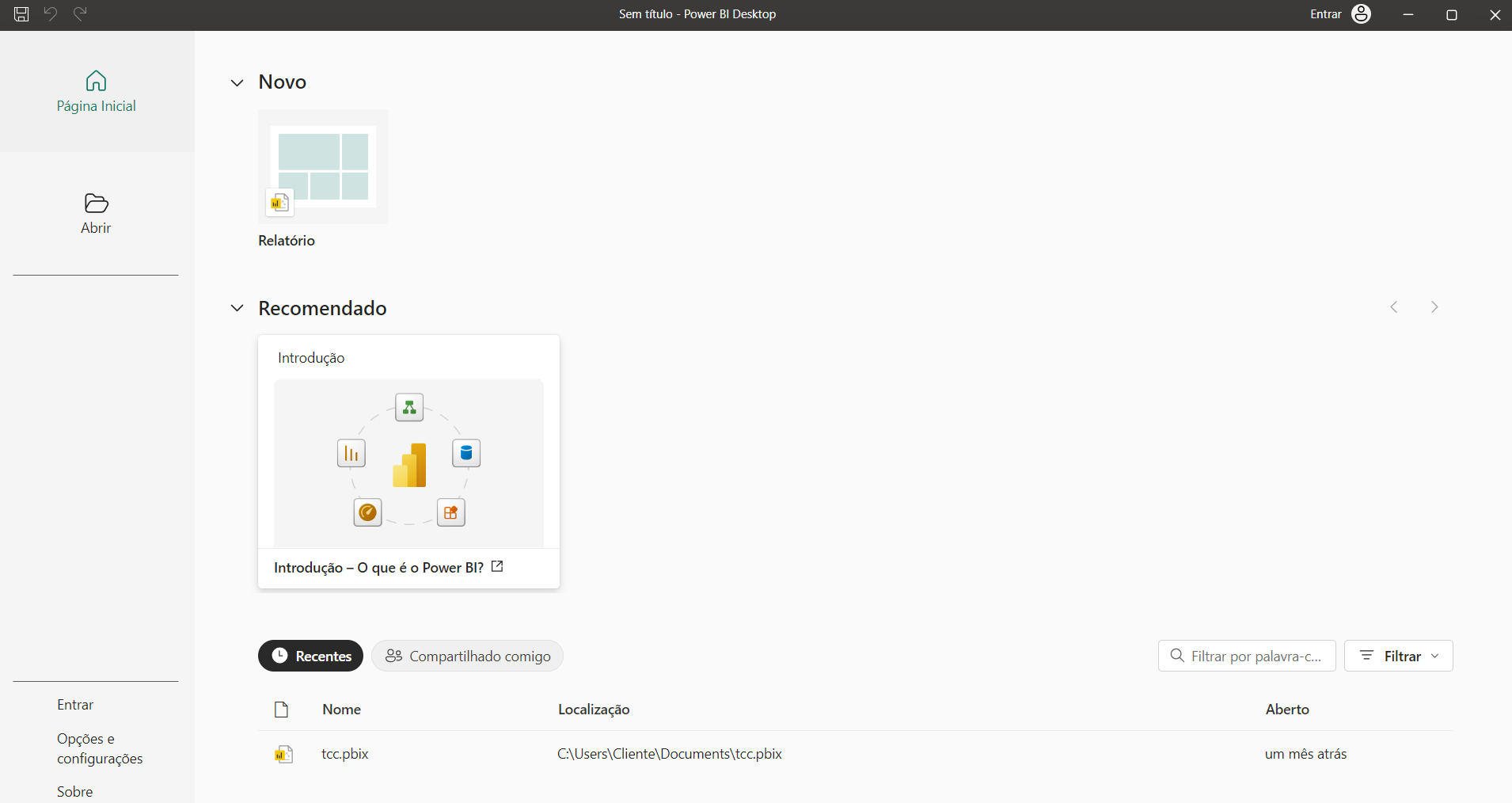
query.write.mode("overwrite").option("mergeSchema", "true").option("path", path\_to\_write).saveAsTable(f"criminal\_gold.Dim\_Municipios\_Brasil")

# COMMAND ----------

## 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.

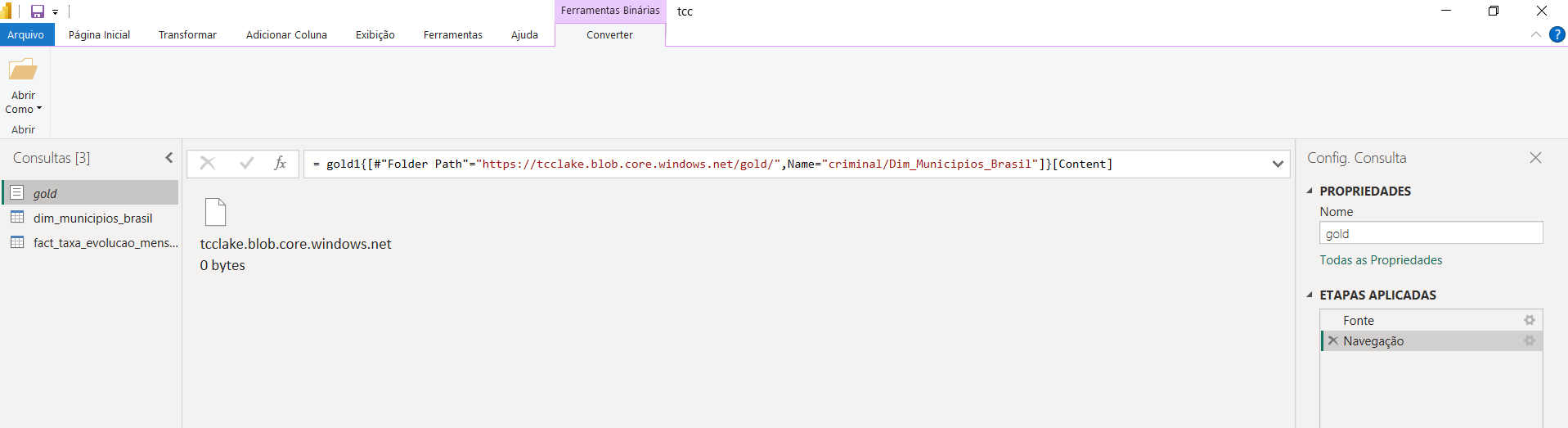
**Figura 14**: Screenshot – Microsoft Power BI



Fonte: Autor

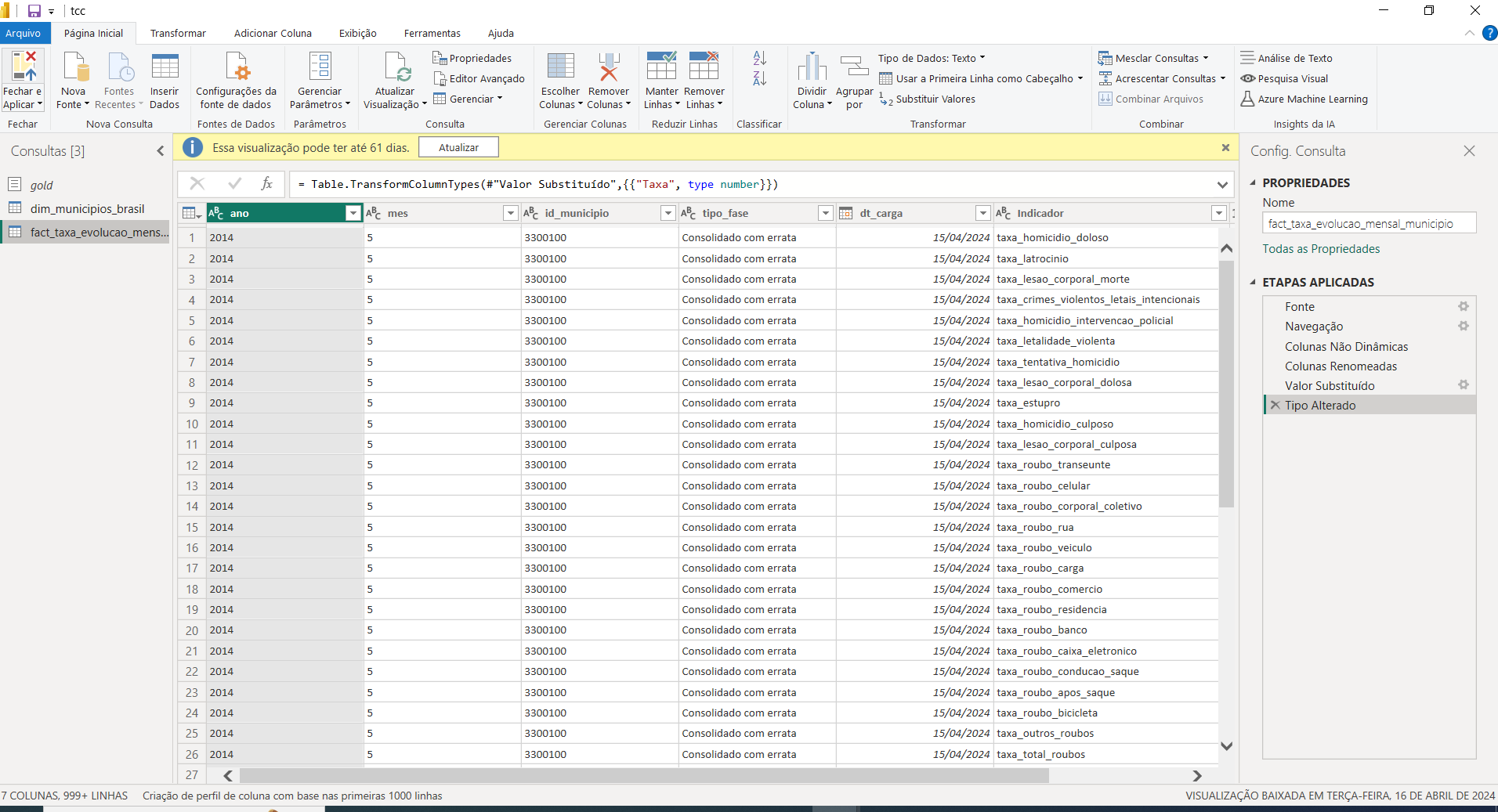
Foi utilizado uma conexão do Microsoft Power BI, diretamente com o DataBrcik Catalog. Conforme imagem abaixo.

Aqui podemos observar a conexão e importação dos dados diretamente da camada Gold.

**Figura 15**: Screenshot – Power Bi x DataBricks

Fonte: Autor

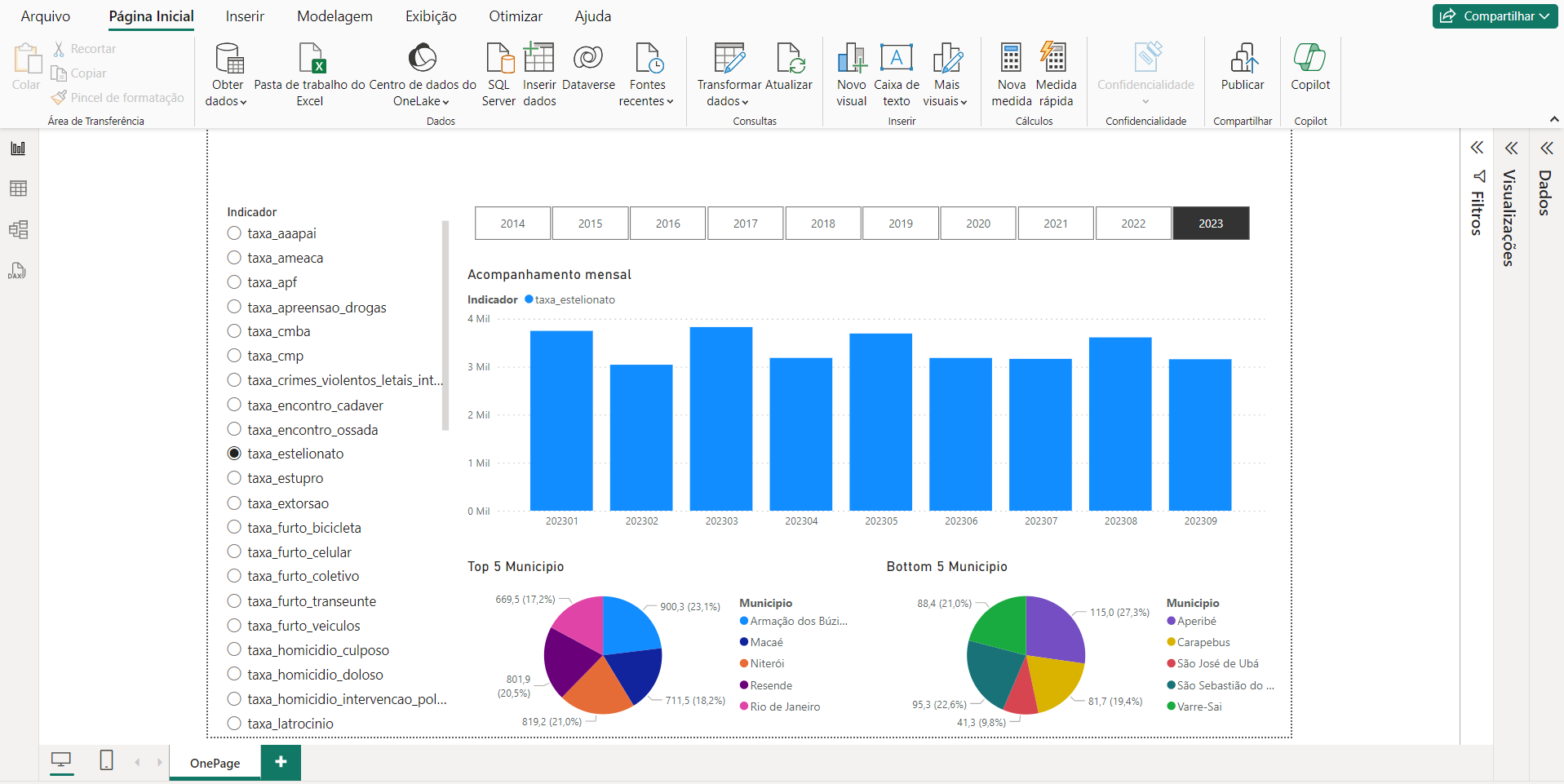
**Figura 16**: Screenshot – Microsoft Power BI – Fact\_taxa\_evolucao\_mensal\_municipio



Fonte: Autor

Nesse Dashboard foram criados alguns filtros por exemplo filtro de “Indicadores”,”Anos. Foi criado um painel inicial para demonstrar os indicadores de forma temporal.

**Figura 17**: Screenshot – Dashboard gerado Power BI



Fonte: Autor

## Apresentação dos Resultados

Os resultados evidenciados neste estudo corroboram importantes discussões no que se refere à criação de estratégias e condições apropriadas, para o mepeamento e implantação de policiamento preventido, tendo uma visão dos pontos que há real necessidade de investir e prevenir.

Com uma navegação rapida e entuitiva, torna-se uma grande ferramenta de apoio a tomada de decisão.

## Links

**Link para o vídeo:**

[**https://youtu.be/bQsmo4wr6S0**](https://youtu.be/bQsmo4wr6S0)

**Link para o repositório Github:**

<https://github.com/sepulvedaleobi/tccpuc>

1. **Referencias**

DataBrciks Disponível em: <[https://community.databricks.com/>.](https://community.databricks.com/%3e.)

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

Azure Disponível em: < https://azure.microsoft.com/>.

# APÊNDICE

## Orquestração/Scripts

Notebooks do processo

**0001 - Ingestion - taxa\_evolucao\_mensal\_municipio.csv**

# Databricks notebook source

# MAGIC %md

# MAGIC ### Leitura do arquivo taxa\_evolucao\_mensal\_municipio.csv

# COMMAND ----------

csv = spark.read.format("csv").option("header", "true").option("sep", ",").option("encoding", "ISO-8859-1").load(f'/mnt/raw/taxa\_evolucao\_mensal\_municipio.csv').createOrReplaceTempView('taxa\_evolucao\_mensal\_municipio')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Variaves de controle de carga

# COMMAND ----------

from datetime import datetime, timedelta

current\_date = datetime.now()

max\_yearmonth\_date = current\_date

max\_yearmonth = max\_yearmonth\_date.strftime('%Y%m')

min\_yearmonth\_date = current\_date - timedelta(days=current\_date.day - 1) - timedelta(weeks=12)

min\_yearmonth = min\_yearmonth\_date.strftime('%Y%m')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Data preparation

# COMMAND ----------

query = spark.sql(f'''

Select

\*,

current\_Date() as dt\_carga

From taxa\_evolucao\_mensal\_municipio

Where 1=1

-- and Concat(ano, LPAD(mes, 2, '0')) >= {min\_yearmonth} Liberar quando houver mais dados na base

-- and Concat(ano, LPAD(mes, 2, '0')) <= {max\_yearmonth} Liberar quando houver mais dados na base

''')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Salvando os dados na camada bronze

# COMMAND ----------

path\_to\_write = '/mnt/bronze/criminal/Tb\_taxa\_evolucao\_mensal\_municipio'

query.write.mode("overwrite").option("mergeSchema", "true").option("path", path\_to\_write).saveAsTable(f"criminal\_bronze.Tb\_taxa\_evolucao\_mensal\_municipio")

**0002 - Ingestion - taxa\_evolucao\_mensal\_municipio\_Bronze\_To\_Silver**

# Databricks notebook source

# MAGIC %md

# MAGIC ### Variaves de controle de carga

# COMMAND ----------

from pyspark.sql import SparkSession

from datetime import datetime, timedelta

# COMMAND ----------

max\_yearmonth = spark.sql("""

Select Distinct Max(Concat(ano, LPAD(mes, 2, '0'))) as Year\_Month

From criminal\_bronze.Tb\_taxa\_evolucao\_mensal\_municipio

""").collect()[0][0]

min\_yearmonth = spark.sql('''Select Min(Concat(ano, LPAD(mes, 2, '0'))) as Year\_Month

From criminal\_bronze.Tb\_taxa\_evolucao\_mensal\_municipio

''').collect()[0][0]

# COMMAND ----------

# MAGIC %md

# MAGIC ### Limpeza do range de dado que será carregado

# COMMAND ----------

result = spark.sql(f"""Delete From criminal\_silver.Tb\_taxa\_evolucao\_mensal\_municipio

Where 1=1

and CONCAT(ano, LPAD(mes, 2, '0')) >= {min\_yearmonth}

and CONCAT(ano, LPAD(mes, 2, '0')) <= {max\_yearmonth}

"""

)

# COMMAND ----------

# MAGIC %md

# MAGIC ### Data preparation

# COMMAND ----------

query = spark.sql(f'''

Select

ano,

mes,

id\_municipio,

regiao,

taxa\_homicidio\_doloso,

taxa\_latrocinio,

taxa\_lesao\_corporal\_morte,

taxa\_crimes\_violentos\_letais\_intencionais,

taxa\_homicidio\_intervencao\_policial,

taxa\_letalidade\_violenta,

taxa\_tentativa\_homicidio,

taxa\_lesao\_corporal\_dolosa,

taxa\_estupro,

taxa\_homicidio\_culposo,

taxa\_lesao\_corporal\_culposa,

taxa\_roubo\_transeunte,

taxa\_roubo\_celular,

taxa\_roubo\_corporal\_coletivo,

taxa\_roubo\_rua,

taxa\_roubo\_veiculo,

taxa\_roubo\_carga,

taxa\_roubo\_comercio,

taxa\_roubo\_residencia,

taxa\_roubo\_banco,

taxa\_roubo\_caixa\_eletronico,

taxa\_roubo\_conducao\_saque,

taxa\_roubo\_apos\_saque,

taxa\_roubo\_bicicleta,

taxa\_outros\_roubos,

taxa\_total\_roubos,

taxa\_furto\_veiculos,

taxa\_furto\_transeunte,

taxa\_furto\_coletivo,

taxa\_furto\_celular,

taxa\_furto\_bicicleta,

taxa\_outros\_furtos,

taxa\_total\_furtos,

taxa\_sequestro,

taxa\_extorsao,

taxa\_sequestro\_relampago,

taxa\_estelionato,

taxa\_apreensao\_drogas,

taxa\_registro\_posse\_drogas,

taxa\_registro\_trafico\_drogas,

taxa\_registro\_apreensao\_drogas\_sem\_autor,

taxa\_registro\_veiculo\_recuperado,

taxa\_apf,

taxa\_aaapai,

taxa\_cmp,

taxa\_cmba,

taxa\_ameaca,

taxa\_pessoas\_desaparecidas,

taxa\_encontro\_cadaver,

taxa\_encontro\_ossada,

taxa\_policial\_militar\_morto\_servico,

taxa\_policial\_civil\_morto\_servico,

taxa\_registro\_ocorrencia,

tipo\_fase,

current\_Date() as dt\_carga

From criminal\_bronze.Tb\_taxa\_evolucao\_mensal\_municipio

Where 1=1

and CONCAT(ano, LPAD(mes, 2, '0')) >= '{min\_yearmonth}'

and CONCAT(ano, LPAD(mes, 2, '0')) <= '{max\_yearmonth}'

''')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Salvando os dados na camada bronze

# COMMAND ----------

path\_to\_write = '/mnt/silver/criminal/Tb\_taxa\_evolucao\_mensal\_municipio'

query.write.mode("append").option("mergeSchema", "true").option("path", path\_to\_write).saveAsTable(f"criminal\_silver.Tb\_taxa\_evolucao\_mensal\_municipio")

# COMMAND ----------

**0003 - Ingestion - taxa\_evolucao\_mensal\_municipio\_Silver\_To\_Gold**

# Databricks notebook source

from pyspark.sql import SparkSession

from datetime import datetime, timedelta

# COMMAND ----------

# MAGIC %md

# MAGIC ### Variaves de controle de carga

# COMMAND ----------

max\_yearmonth = spark.sql("""

Select Distinct Max(Concat(ano, LPAD(mes, 2, '0'))) as Year\_Month

From criminal\_bronze.Tb\_taxa\_evolucao\_mensal\_municipio

""").collect()[0][0]

min\_yearmonth = spark.sql('''Select Min(Concat(ano, LPAD(mes, 2, '0'))) as Year\_Month

From criminal\_bronze.Tb\_taxa\_evolucao\_mensal\_municipio

''').collect()[0][0]

# COMMAND ----------

# MAGIC %md

# MAGIC ### Limpeza do range de dado que será carregado

# COMMAND ----------

result = spark.sql(f"""Delete From criminal\_gold.Fact\_taxa\_evolucao\_mensal\_municipio

Where 1=1

and CONCAT(ano, LPAD(mes, 2, '0')) >= {min\_yearmonth}

and CONCAT(ano, LPAD(mes, 2, '0')) <= {max\_yearmonth}

"""

)

# COMMAND ----------

# MAGIC %md

# MAGIC ### Data preparation

# COMMAND ----------

query = spark.sql(f'''

Select

ano,

mes,

id\_municipio,

taxa\_homicidio\_doloso,

taxa\_latrocinio,

taxa\_lesao\_corporal\_morte,

taxa\_crimes\_violentos\_letais\_intencionais,

taxa\_homicidio\_intervencao\_policial,

taxa\_letalidade\_violenta,

taxa\_tentativa\_homicidio,

taxa\_lesao\_corporal\_dolosa,

taxa\_estupro,

taxa\_homicidio\_culposo,

taxa\_lesao\_corporal\_culposa,

taxa\_roubo\_transeunte,

taxa\_roubo\_celular,

taxa\_roubo\_corporal\_coletivo,

taxa\_roubo\_rua,

taxa\_roubo\_veiculo,

taxa\_roubo\_carga,

taxa\_roubo\_comercio,

taxa\_roubo\_residencia,

taxa\_roubo\_banco,

taxa\_roubo\_caixa\_eletronico,

taxa\_roubo\_conducao\_saque,

taxa\_roubo\_apos\_saque,

taxa\_roubo\_bicicleta,

taxa\_outros\_roubos,

taxa\_total\_roubos,

taxa\_furto\_veiculos,

taxa\_furto\_transeunte,

taxa\_furto\_coletivo,

taxa\_furto\_celular,

taxa\_furto\_bicicleta,

taxa\_outros\_furtos,

taxa\_total\_furtos,

taxa\_sequestro,

taxa\_extorsao,

taxa\_sequestro\_relampago,

taxa\_estelionato,

taxa\_apreensao\_drogas,

taxa\_registro\_posse\_drogas,

taxa\_registro\_trafico\_drogas,

taxa\_registro\_apreensao\_drogas\_sem\_autor,

taxa\_registro\_veiculo\_recuperado,

taxa\_apf,

taxa\_aaapai,

taxa\_cmp,

taxa\_cmba,

taxa\_ameaca,

taxa\_pessoas\_desaparecidas,

taxa\_encontro\_cadaver,

taxa\_encontro\_ossada,

taxa\_policial\_militar\_morto\_servico,

taxa\_policial\_civil\_morto\_servico,

taxa\_registro\_ocorrencia,

tipo\_fase,

current\_Date() as dt\_carga

From criminal\_silver.Tb\_taxa\_evolucao\_mensal\_municipio

Where 1=1

and CONCAT(ano, LPAD(mes, 2, '0')) >= '{min\_yearmonth}'

and CONCAT(ano, LPAD(mes, 2, '0')) <= '{max\_yearmonth}'

''')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Salvando os dados na camada bronze

# COMMAND ----------

path\_to\_write = '/mnt/gold/criminal/Fact\_taxa\_evolucao\_mensal\_municipio'

query.write.mode("overwrite").option("mergeSchema", "true").option("path", path\_to\_write).saveAsTable(f"criminal\_gold.Fact\_taxa\_evolucao\_mensal\_municipio")

# COMMAND ----------

**0004 - Ingestion -municipio**

# Databricks notebook source

# MAGIC %md

# MAGIC ### Leitura do arquivo taxa\_evolucao\_mensal\_municipio.csv

# COMMAND ----------

csv = spark.read.format("csv").option("header", "true").option("sep", ";").option("encoding", "UTF-8").load(f'/mnt/raw/RELATORIO\_DTB\_BRASIL\_MUNICIPIO.csv').createOrReplaceTempView('Municipios\_Brasil')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Data preparation

# COMMAND ----------

query = spark.sql(f'''

Select

UF as UF,

Nome\_UF as Nome\_UF,

`Região Geográfica Intermediária` as Regiao\_Geografica\_Intermediaria,

`Nome Região Geográfica Intermediária` as Nome\_Regiao\_Geografica\_Intermediaria,

`Região Geográfica Imediata` as Regiao\_Geografica\_Imediata,

`Nome Região Geográfica Imediata` as Nome\_Regiao\_Geografica\_Imediata,

`Mesorregião Geográfica` as MesoRegiao\_Geografica,

`Nome\_Mesorregião` as Nome\_MesoRegiao,

`Microrregião Geográfica` as MicroRegiao\_Geografica,

`Nome\_Microrregião` as Nome\_MicroRegiao,

`Município` as Codigo\_Municipio,

`Código Município Completo` as Id\_Municipio,

`Nome\_Município` as Nome\_Municipio,

current\_Date() as Dt\_Carga

From Municipios\_Brasil

Where 1=1

''')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Salvando os dados na camada bronze

# COMMAND ----------

path\_to\_write = '/mnt/bronze/criminal/Tb\_Municipios\_Brasil'

query.write.mode("overwrite").option("mergeSchema", "true").option("path", path\_to\_write).saveAsTable(f"criminal\_bronze.Tb\_Municipios\_Brasil")

# COMMAND ----------

# MAGIC %sql

# MAGIC Select \*

# MAGIC From Delta.`/mnt/bronze/criminal/Tb\_Municipios\_Brasil`

# COMMAND ----------

**0005 - Ingestion - Municipios\_Brasil\_Bronze\_To\_Gold**

# Databricks notebook source

# MAGIC %md

# MAGIC ### Leitura do arquivo taxa\_evolucao\_mensal\_municipio.csv

# COMMAND ----------

result = spark.sql(f"""Delete From criminal\_gold.Dim\_Municipios\_Brasil

Where 1=1

"""

)

# COMMAND ----------

# MAGIC %md

# MAGIC ### Data preparation

# COMMAND ----------

query = spark.sql(f'''

Select

UF,

Nome\_UF,

Regiao\_Geografica\_Intermediaria,

Nome\_Regiao\_Geografica\_Intermediaria,

Regiao\_Geografica\_Imediata,

Nome\_Regiao\_Geografica\_Imediata,

MesoRegiao\_Geografica,

Nome\_MesoRegiao,

MicroRegiao\_Geografica,

Nome\_MicroRegiao,

Codigo\_Municipio,

Id\_Municipio,

Nome\_Municipio,

Dt\_Carga

From criminal\_bronze.Tb\_Municipios\_Brasil

Where 1=1

''')

# COMMAND ----------

# MAGIC %md

# MAGIC ### Salvando os dados na camada bronze

# COMMAND ----------

path\_to\_write = '/mnt/gold/criminal/Dim\_Municipios\_Brasil'

query.write.mode("overwrite").option("mergeSchema", "true").option("path", path\_to\_write).saveAsTable(f"criminal\_gold.Dim\_Municipios\_Brasil")

# COMMAND ----------

Script de Criação das Tabelas

# Databricks notebook source

# MAGIC %md

# MAGIC #### Estruturação dos DataBase HIVE para cada camada

# COMMAND ----------

# MAGIC %sql

# MAGIC Create DataBase if not Exists criminal\_bronze

# COMMAND ----------

# MAGIC %sql

# MAGIC Create DataBase if not Exists criminal\_silver

# COMMAND ----------

# MAGIC %sql

# MAGIC Create DataBase if not Exists criminal\_gold

# COMMAND ----------

# MAGIC %md

# MAGIC #### Criação das tabelas camada Bronze

# COMMAND ----------

# MAGIC %sql

# MAGIC CREATE TABLE criminal\_bronze.Tb\_taxa\_evolucao\_mensal\_municipio (

# MAGIC ano STRING,

# MAGIC mes STRING,

# MAGIC id\_municipio STRING,

# MAGIC regiao STRING,

# MAGIC taxa\_homicidio\_doloso STRING,

# MAGIC taxa\_latrocinio STRING,

# MAGIC taxa\_lesao\_corporal\_morte STRING,

# MAGIC taxa\_crimes\_violentos\_letais\_intencionais STRING,

# MAGIC taxa\_homicidio\_intervencao\_policial STRING,

# MAGIC taxa\_letalidade\_violenta STRING,

# MAGIC taxa\_tentativa\_homicidio STRING,

# MAGIC taxa\_lesao\_corporal\_dolosa STRING,

# MAGIC taxa\_estupro STRING,

# MAGIC taxa\_homicidio\_culposo STRING,

# MAGIC taxa\_lesao\_corporal\_culposa STRING,

# MAGIC taxa\_roubo\_transeunte STRING,

# MAGIC taxa\_roubo\_celular STRING,

# MAGIC taxa\_roubo\_corporal\_coletivo STRING,

# MAGIC taxa\_roubo\_rua STRING,

# MAGIC taxa\_roubo\_veiculo STRING,

# MAGIC taxa\_roubo\_carga STRING,

# MAGIC taxa\_roubo\_comercio STRING,

# MAGIC taxa\_roubo\_residencia STRING,

# MAGIC taxa\_roubo\_banco STRING,

# MAGIC taxa\_roubo\_caixa\_eletronico STRING,

# MAGIC taxa\_roubo\_conducao\_saque STRING,

# MAGIC taxa\_roubo\_apos\_saque STRING,

# MAGIC taxa\_roubo\_bicicleta STRING,

# MAGIC taxa\_outros\_roubos STRING,

# MAGIC taxa\_total\_roubos STRING,

# MAGIC taxa\_furto\_veiculos STRING,

# MAGIC taxa\_furto\_transeunte STRING,

# MAGIC taxa\_furto\_coletivo STRING,

# MAGIC taxa\_furto\_celular STRING,

# MAGIC taxa\_furto\_bicicleta STRING,

# MAGIC taxa\_outros\_furtos STRING,

# MAGIC taxa\_total\_furtos STRING,

# MAGIC taxa\_sequestro STRING,

# MAGIC taxa\_extorsao STRING,

# MAGIC taxa\_sequestro\_relampago STRING,

# MAGIC taxa\_estelionato STRING,

# MAGIC taxa\_apreensao\_drogas STRING,

# MAGIC taxa\_registro\_posse\_drogas STRING,

# MAGIC taxa\_registro\_trafico\_drogas STRING,

# MAGIC taxa\_registro\_apreensao\_drogas\_sem\_autor STRING,

# MAGIC taxa\_registro\_veiculo\_recuperado STRING,

# MAGIC taxa\_apf STRING,

# MAGIC taxa\_aaapai STRING,

# MAGIC taxa\_cmp STRING,

# MAGIC taxa\_cmba STRING,

# MAGIC taxa\_ameaca STRING,

# MAGIC taxa\_pessoas\_desaparecidas STRING,

# MAGIC taxa\_encontro\_cadaver STRING,

# MAGIC taxa\_encontro\_ossada STRING,

# MAGIC taxa\_policial\_militar\_morto\_servico STRING,

# MAGIC taxa\_policial\_civil\_morto\_servico STRING,

# MAGIC taxa\_registro\_ocorrencia STRING,

# MAGIC tipo\_fase STRING,

# MAGIC dt\_carga DATE)

# MAGIC USING delta

# MAGIC LOCATION 'dbfs:/mnt/bronze/criminal/Tb\_taxa\_evolucao\_mensal\_municipio'

# MAGIC TBLPROPERTIES (

# MAGIC 'delta.minReaderVersion' = '1',

# MAGIC 'delta.minWriterVersion' = '2')

# MAGIC

# COMMAND ----------

# MAGIC %sql

# MAGIC CREATE TABLE criminal\_bronze.tb\_municipios\_brasil (

# MAGIC UF STRING,

# MAGIC Nome\_UF STRING,

# MAGIC Regiao\_Geografica\_Intermediaria STRING,

# MAGIC Nome\_Regiao\_Geografica\_Intermediaria STRING,

# MAGIC Regiao\_Geografica\_Imediata STRING,

# MAGIC Nome\_Regiao\_Geografica\_Imediata STRING,

# MAGIC MesoRegiao\_Geografica STRING,

# MAGIC Nome\_MesoRegiao STRING,

# MAGIC MicroRegiao\_Geografica STRING,

# MAGIC Nome\_MicroRegiao STRING,

# MAGIC Codigo\_Municipio STRING,

# MAGIC Id\_Municipio STRING,

# MAGIC Nome\_Municipio STRING,

# MAGIC Dt\_Carga DATE)

# MAGIC USING delta

# MAGIC LOCATION 'dbfs:/mnt/bronze/criminal/Tb\_Municipios\_Brasil'

# MAGIC TBLPROPERTIES (

# MAGIC 'delta.minReaderVersion' = '1',

# MAGIC 'delta.minWriterVersion' = '2')

# MAGIC

# COMMAND ----------

# MAGIC %md

# MAGIC #### Criação das tabelas camada Silver

# COMMAND ----------

# MAGIC %sql

# MAGIC CREATE TABLE criminal\_silver.Tb\_taxa\_evolucao\_mensal\_municipio (

# MAGIC ano STRING,

# MAGIC mes STRING,

# MAGIC id\_municipio STRING,

# MAGIC regiao STRING,

# MAGIC taxa\_homicidio\_doloso STRING,

# MAGIC taxa\_latrocinio STRING,

# MAGIC taxa\_lesao\_corporal\_morte STRING,

# MAGIC taxa\_crimes\_violentos\_letais\_intencionais STRING,

# MAGIC taxa\_homicidio\_intervencao\_policial STRING,

# MAGIC taxa\_letalidade\_violenta STRING,

# MAGIC taxa\_tentativa\_homicidio STRING,

# MAGIC taxa\_lesao\_corporal\_dolosa STRING,

# MAGIC taxa\_estupro STRING,

# MAGIC taxa\_homicidio\_culposo STRING,

# MAGIC taxa\_lesao\_corporal\_culposa STRING,

# MAGIC taxa\_roubo\_transeunte STRING,

# MAGIC taxa\_roubo\_celular STRING,

# MAGIC taxa\_roubo\_corporal\_coletivo STRING,

# MAGIC taxa\_roubo\_rua STRING,

# MAGIC taxa\_roubo\_veiculo STRING,

# MAGIC taxa\_roubo\_carga STRING,

# MAGIC taxa\_roubo\_comercio STRING,

# MAGIC taxa\_roubo\_residencia STRING,

# MAGIC taxa\_roubo\_banco STRING,

# MAGIC taxa\_roubo\_caixa\_eletronico STRING,

# MAGIC taxa\_roubo\_conducao\_saque STRING,

# MAGIC taxa\_roubo\_apos\_saque STRING,

# MAGIC taxa\_roubo\_bicicleta STRING,

# MAGIC taxa\_outros\_roubos STRING,

# MAGIC taxa\_total\_roubos STRING,

# MAGIC taxa\_furto\_veiculos STRING,

# MAGIC taxa\_furto\_transeunte STRING,

# MAGIC taxa\_furto\_coletivo STRING,

# MAGIC taxa\_furto\_celular STRING,

# MAGIC taxa\_furto\_bicicleta STRING,

# MAGIC taxa\_outros\_furtos STRING,

# MAGIC taxa\_total\_furtos STRING,

# MAGIC taxa\_sequestro STRING,

# MAGIC taxa\_extorsao STRING,

# MAGIC taxa\_sequestro\_relampago STRING,

# MAGIC taxa\_estelionato STRING,

# MAGIC taxa\_apreensao\_drogas STRING,

# MAGIC taxa\_registro\_posse\_drogas STRING,

# MAGIC taxa\_registro\_trafico\_drogas STRING,

# MAGIC taxa\_registro\_apreensao\_drogas\_sem\_autor STRING,

# MAGIC taxa\_registro\_veiculo\_recuperado STRING,

# MAGIC taxa\_apf STRING,

# MAGIC taxa\_aaapai STRING,

# MAGIC taxa\_cmp STRING,

# MAGIC taxa\_cmba STRING,

# MAGIC taxa\_ameaca STRING,

# MAGIC taxa\_pessoas\_desaparecidas STRING,

# MAGIC taxa\_encontro\_cadaver STRING,

# MAGIC taxa\_encontro\_ossada STRING,

# MAGIC taxa\_policial\_militar\_morto\_servico STRING,

# MAGIC taxa\_policial\_civil\_morto\_servico STRING,

# MAGIC taxa\_registro\_ocorrencia STRING,

# MAGIC tipo\_fase STRING,

# MAGIC dt\_carga DATE)

# MAGIC USING delta

# MAGIC LOCATION 'dbfs:/mnt/silver/criminal/Tb\_taxa\_evolucao\_mensal\_municipio'

# MAGIC TBLPROPERTIES (

# MAGIC 'delta.minReaderVersion' = '1',

# MAGIC 'delta.minWriterVersion' = '2')

# MAGIC

# COMMAND ----------

# MAGIC %md

# MAGIC #### Criação das tabelas camada Gold

# COMMAND ----------

# MAGIC %sql

# MAGIC CREATE TABLE criminal\_gold.Fact\_taxa\_evolucao\_mensal\_municipio (

# MAGIC ano STRING,

# MAGIC mes STRING,

# MAGIC id\_municipio STRING,

# MAGIC taxa\_homicidio\_doloso STRING,

# MAGIC taxa\_latrocinio STRING,

# MAGIC taxa\_lesao\_corporal\_morte STRING,

# MAGIC taxa\_crimes\_violentos\_letais\_intencionais STRING,

# MAGIC taxa\_homicidio\_intervencao\_policial STRING,

# MAGIC taxa\_letalidade\_violenta STRING,

# MAGIC taxa\_tentativa\_homicidio STRING,

# MAGIC taxa\_lesao\_corporal\_dolosa STRING,

# MAGIC taxa\_estupro STRING,

# MAGIC taxa\_homicidio\_culposo STRING,

# MAGIC taxa\_lesao\_corporal\_culposa STRING,

# MAGIC taxa\_roubo\_transeunte STRING,

# MAGIC taxa\_roubo\_celular STRING,

# MAGIC taxa\_roubo\_corporal\_coletivo STRING,

# MAGIC taxa\_roubo\_rua STRING,

# MAGIC taxa\_roubo\_veiculo STRING,

# MAGIC taxa\_roubo\_carga STRING,

# MAGIC taxa\_roubo\_comercio STRING,

# MAGIC taxa\_roubo\_residencia STRING,

# MAGIC taxa\_roubo\_banco STRING,

# MAGIC taxa\_roubo\_caixa\_eletronico STRING,

# MAGIC taxa\_roubo\_conducao\_saque STRING,

# MAGIC taxa\_roubo\_apos\_saque STRING,

# MAGIC taxa\_roubo\_bicicleta STRING,

# MAGIC taxa\_outros\_roubos STRING,

# MAGIC taxa\_total\_roubos STRING,

# MAGIC taxa\_furto\_veiculos STRING,

# MAGIC taxa\_furto\_transeunte STRING,

# MAGIC taxa\_furto\_coletivo STRING,

# MAGIC taxa\_furto\_celular STRING,

# MAGIC taxa\_furto\_bicicleta STRING,

# MAGIC taxa\_outros\_furtos STRING,

# MAGIC taxa\_total\_furtos STRING,

# MAGIC taxa\_sequestro STRING,

# MAGIC taxa\_extorsao STRING,

# MAGIC taxa\_sequestro\_relampago STRING,

# MAGIC taxa\_estelionato STRING,

# MAGIC taxa\_apreensao\_drogas STRING,

# MAGIC taxa\_registro\_posse\_drogas STRING,

# MAGIC taxa\_registro\_trafico\_drogas STRING,

# MAGIC taxa\_registro\_apreensao\_drogas\_sem\_autor STRING,

# MAGIC taxa\_registro\_veiculo\_recuperado STRING,

# MAGIC taxa\_apf STRING,

# MAGIC taxa\_aaapai STRING,

# MAGIC taxa\_cmp STRING,

# MAGIC taxa\_cmba STRING,

# MAGIC taxa\_ameaca STRING,

# MAGIC taxa\_pessoas\_desaparecidas STRING,

# MAGIC taxa\_encontro\_cadaver STRING,

# MAGIC taxa\_encontro\_ossada STRING,

# MAGIC taxa\_policial\_militar\_morto\_servico STRING,

# MAGIC taxa\_policial\_civil\_morto\_servico STRING,

# MAGIC taxa\_registro\_ocorrencia STRING,

# MAGIC tipo\_fase STRING,

# MAGIC dt\_carga DATE)

# MAGIC USING delta

# MAGIC LOCATION 'dbfs:/mnt/gold/criminal/Fact\_taxa\_evolucao\_mensal\_municipio'

# MAGIC TBLPROPERTIES (

# MAGIC 'delta.minReaderVersion' = '1',

# MAGIC 'delta.minWriterVersion' = '2')

# MAGIC

# COMMAND ----------

# MAGIC %sql

# MAGIC CREATE TABLE criminal\_gold.dim\_municipios\_brasil (

# MAGIC UF STRING,

# MAGIC Nome\_UF STRING,

# MAGIC Regiao\_Geografica\_Intermediaria STRING,

# MAGIC Nome\_Regiao\_Geografica\_Intermediaria STRING,

# MAGIC Regiao\_Geografica\_Imediata STRING,

# MAGIC Nome\_Regiao\_Geografica\_Imediata STRING,

# MAGIC MesoRegiao\_Geografica STRING,

# MAGIC Nome\_MesoRegiao STRING,

# MAGIC MicroRegiao\_Geografica STRING,

# MAGIC Nome\_MicroRegiao STRING,

# MAGIC Codigo\_Municipio STRING,

# MAGIC Id\_Municipio STRING,

# MAGIC Nome\_Municipio STRING,

# MAGIC Dt\_Carga DATE)

# MAGIC USING delta

# MAGIC LOCATION 'dbfs:/mnt/gold/criminal/Dim\_Municipios\_Brasil'

# MAGIC TBLPROPERTIES (

# MAGIC 'delta.minReaderVersion' = '1',

# MAGIC 'delta.minWriterVersion' = '2')

# MAGIC