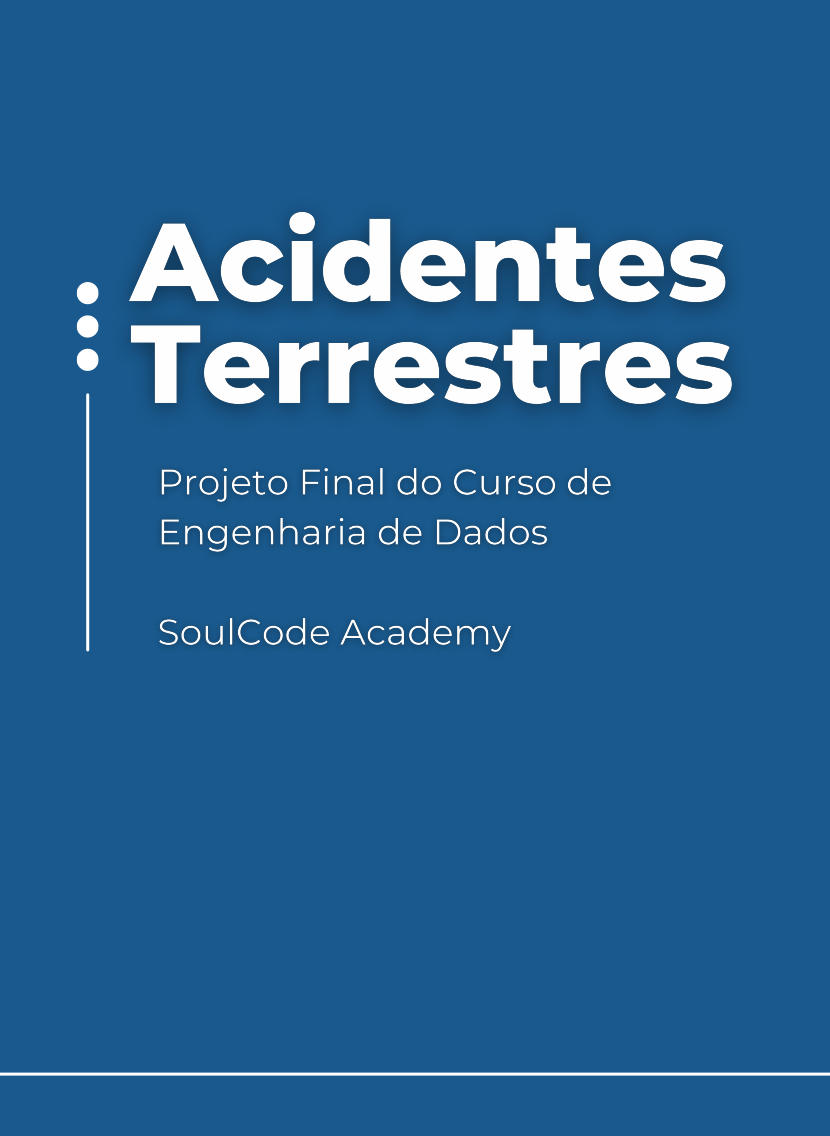
****

**Projeto Final**

**SoulCode Academy**

**Trabalho Final do Curso de Engenharia de Dados**

**Turma BC-12**

Projeto desenvolvido pelos alunos:

Alyne Cristina

Eveline Marques

Gustavo Santoro

Matheus Reis

**Requisitos**

Data sets públicos que abrangem o tema Acidentes Terrestres que contenham informações sobre acidentes, as rodovias onde mais acontecem e dos quais seja possível analisar por ano, rodovia e estados as características dos acidentes, suas causas e número de pessoas envolvidas.

Deverão ser encontrados e tratados no mínimo dois data sets sobre o assunto em questão. Seguindo alguns passos como:

* Os dados devem ser de formatos diferentes;
* Deve-se usar pandas, PySpark e Spark SQL para transformações e consultas;
* Armazenamento dos dados brutos em na Cloud SQL com MySql;
* Os dados devem estar disponíveis na Google Cloud Storage e BigQuery;
* Os dados tratados devem ser enviados para o MongoDB;

**Origem dos Dados**

O dados utilizados no processo de ETL desse documento foram colhidos na plataforma de [Dados Abertos da Polícia Rodoviária Federal](https://www.gov.br/prf/pt-br/acesso-a-informacao/dados-abertos). Foram utilizados os dados sobre Acidentes no período de 2021 à 2017 disponibilizados em arquivos separados por ‘Agrupados por Ocorrências’ e ‘Agrupados por Pessoas’. Os arquivos estão em formato ‘zip’, divididos por ano em que as ocorrências foram registradas.

**Destino dos Dados**

Os arquivos foram armazenados na Google Cloud Storage (Datalake). Para isso foi criada a uma bucket com nome ‘acidentes\_terrestres’ e a pasta ‘dados\_brutos’.

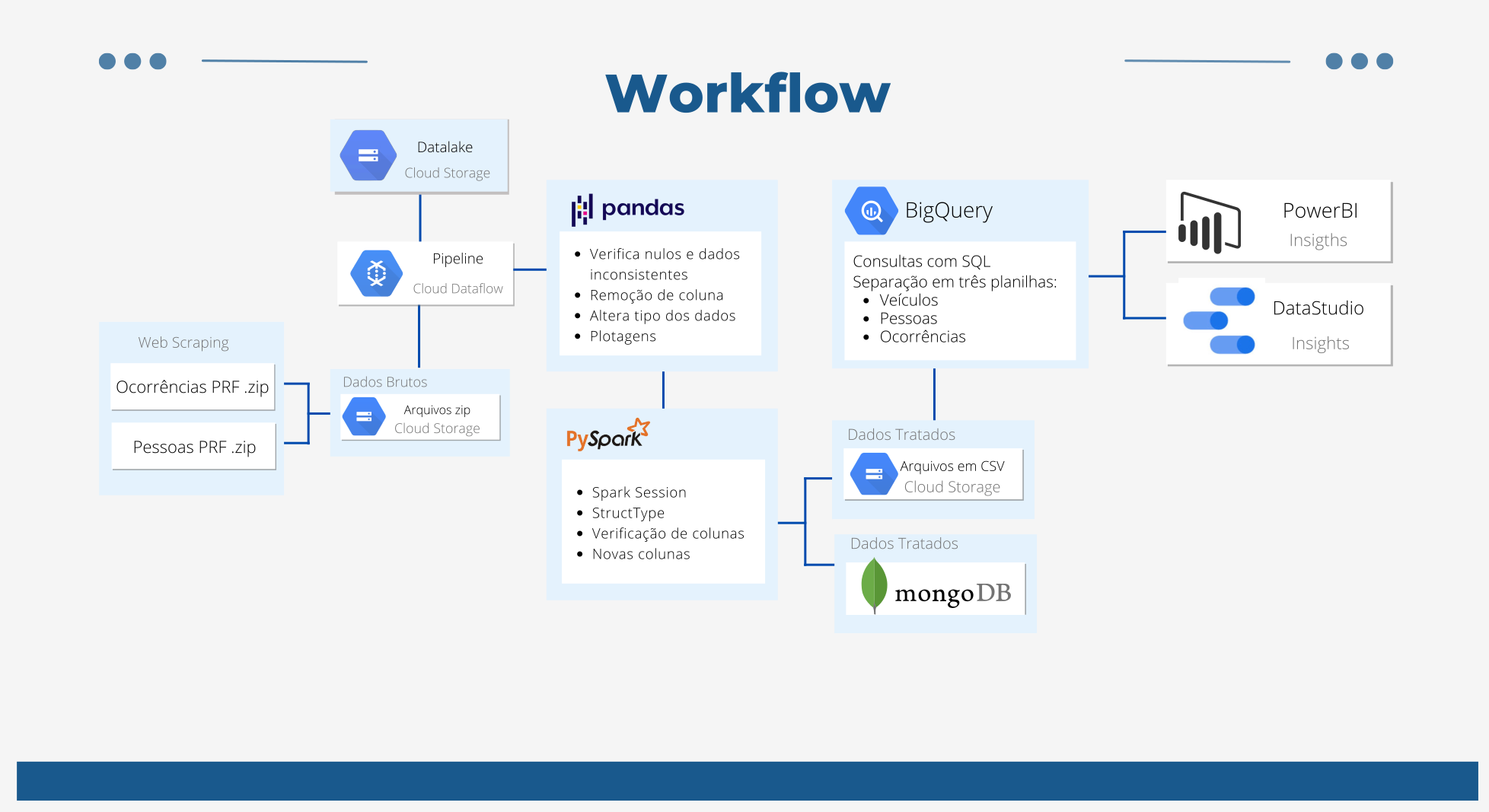
Apos os tratamentos feitos com pandas e PySpark os dados foram enviados para a BigQuery e para MongoDB. O dados tratados e normalizados foram armazenados no Datalake citado na pasta ‘dados\_tratados.’

**Previsão de Gastos de acordo com a GCP**

Estimativa de custo de acordo com a Google Cloud

Link calculadora: **[https://cloud.google.com/products/calculator#id=e468548e-326a-44ae-a7ab-6b81cbbb4f6a](https://cloud.google.com/products/calculator" \l "id=e468548e-326a-44ae-a7ab-6b81cbbb4f6a)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Your Estimated Bill Projeto FinalEstimated Monthly Cost: USD 22.74 (R$ 102.00)  |  |  |  |  | | --- | --- | --- | --- | | 1x Standard Storage | Standard Storage | 5 GiB | USD 0.18 | | info\_ocorrencias | BigQuery | 78.55 GiB | USD 1.37 | | info\_pesspas | BigQuery | 40 GiB | USD 0.00 | | info\_veiculos | BigQuery | 40.97 GiB | USD 0.00 | | Dataproc | projetofinal | 48 | USD 0.48 | | 1 x Dataproc master node | n1-standard-4 | 6 total hours per month | USD 2.18 | | 1 x Dataproc worker nodes | n1-standard-4 | 6 total hours per month | USD 2.18 | | 1x | Persistent Disk - Dataproc | 8.219178082191782 GiB | USD 0.49 | | Dataflow | 1 x n1-standard-1 workers in Batch Mode | 1 | USD 0.11 | | db-standard-1 | 10 GB | 130.35714285714283 total hours per month | USD 15.75 | | Total Estimated Monthly Cost | | | USD 22.74 | |



**Importação/Extração dos dados**

A extração dos dados se deu através de um algoritmo, através de bibliotecas de raspagem de dados, agrega todos os elementos HTML do site da PRF em um dicionário, onde é possível extrair os arquivos compactados.

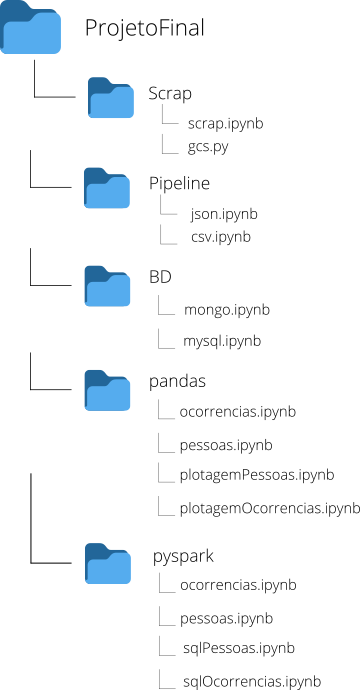
**Ferramentas Utilizadas**

Foram utilizados pandas e PySpark para o tratamento, normalização dos dados e remoção de colunas e dados inconsistentes e plotagem de gráficos com análises iniciais.

Com SparkSql foram feitas buscas que guiaram a elaboração dos relatórios e insights para análise posterior no DataStudio e PowerBi. No BigQuery foram disponibilizados os dados tratados.

Para visualizar os relatórios acesse os links para o [PowerBI](https://app.powerbi.com/view?r=eyJrIjoiYTI1Y2ExZWUtOGUyYi00YjZhLTlhMTQtNzA4NzY0ZGU5ODg0IiwidCI6ImQwZmYyY2NkLTI5MTItNDFhMS05NmM4LTA3NTU3NTg1OTkyYiJ9) e [DataStudio](https://datastudio.google.com/reporting/bf781d54-f1d6-4983-9447-6bccf32cecbf/page/p_07ah5vnltc)

**Arquivos**



**Dicionário dos Dados**

*Acidentes Agrupados por ocorrências*

|  |  |
| --- | --- |
| **NOME DA VARIÁVEL** | **DESCRIÇÃO** |
| *id* | Variável com valores numéricos, representando o  identificador do acidente. |
| *data\_inversa* | Data da ocorrência no formato dd/mm/aaaa. |
| *dia\_semana* | Dia da semana da ocorrência. Ex.: Segunda, Terça, etc. |
| *horario* | Horário da ocorrência no formato hh:mm:ss. |
| *uf* | Unidade da Federação. Ex.: MG, PE, DF, etc. |
| *br* | Variável com valores numéricos, representando o  identificador da BR do acidente. |
| *km* | Identificação do quilômetro onde ocorreu o acidente, com valor mínimo de 0,1 km e com a casa decimal separada por ponto. |
| *municipio* | Nome do município de ocorrência do acidente |
| *causa\_acidente* | Identificação da causa principal do acidente. Neste conjunto de dados são excluídos os acidentes com a  variável causa principal igual a “Não”. |
| *tipo\_acidente* | Identificação do tipo de acidente. Ex.: Colisão frontal, Saída de pista, etc. Neste conjunto de dados são excluídos os tipos de acidentes com ordem maior ou igual a dois. A ordem do acidente demonstra a sequência cronológica dos  tipos presentes na mesma ocorrência. |
| *classificação\_acidente* | Classificação quanto à gravidade do acidente: Sem Vítimas, Com Vítimas Feridas, Com Vítimas Fatais e  Ignorado. |
| *fase\_dia* | Fase do dia no momento do acidente. Ex. Amanhecer,  Pleno dia, etc. |
| *sentido\_via* | Sentido da via considerando o ponto de colisão:  Crescente e decrescente. |
| *condição\_meteorologica* | Condição meteorológica no momento do acidente: Céu  claro, chuva, vento, etc. |
| *tipo\_pista* | Tipo da pista considerando a quantidade de faixas: Dupla,  simples ou múltipla. |
| *tracado\_via* | Descrição do traçado da via. |
| *uso\_solo* | Descrição sobre as características do local do acidente:  Urbano=Sim;Rural=Não. |
| *latitude* | Latitude do local do acidente em formato geodésico decimal. |
| *longitude* | Longitude do local do acidente em formato geodésico decimal. |

|  |  |
| --- | --- |
| *pessoas* | Total de pessoas envolvidas na ocorrência. |
| *mortos* | Total de pessoas mortas envolvidas na ocorrência. |
| *feridos\_leves* | Total de pessoas com ferimentos leves envolvidas na ocorrência. |
| *feridos\_graves* | Total de pessoas com ferimentos graves envolvidas na ocorrência. |
| *feridos* | Total de pessoas feridas envolvidas na ocorrência (é a  soma dos feridos leves com os graves). |
| *ilesos* | Total de pessoas ilesas envolvidas na ocorrência. |
| *ignorados* | Total de pessoas envolvidas na ocorrência e que não se  soube o estado físico. |
| *veiculos* | Total de veículos envolvidos na ocorrência. |

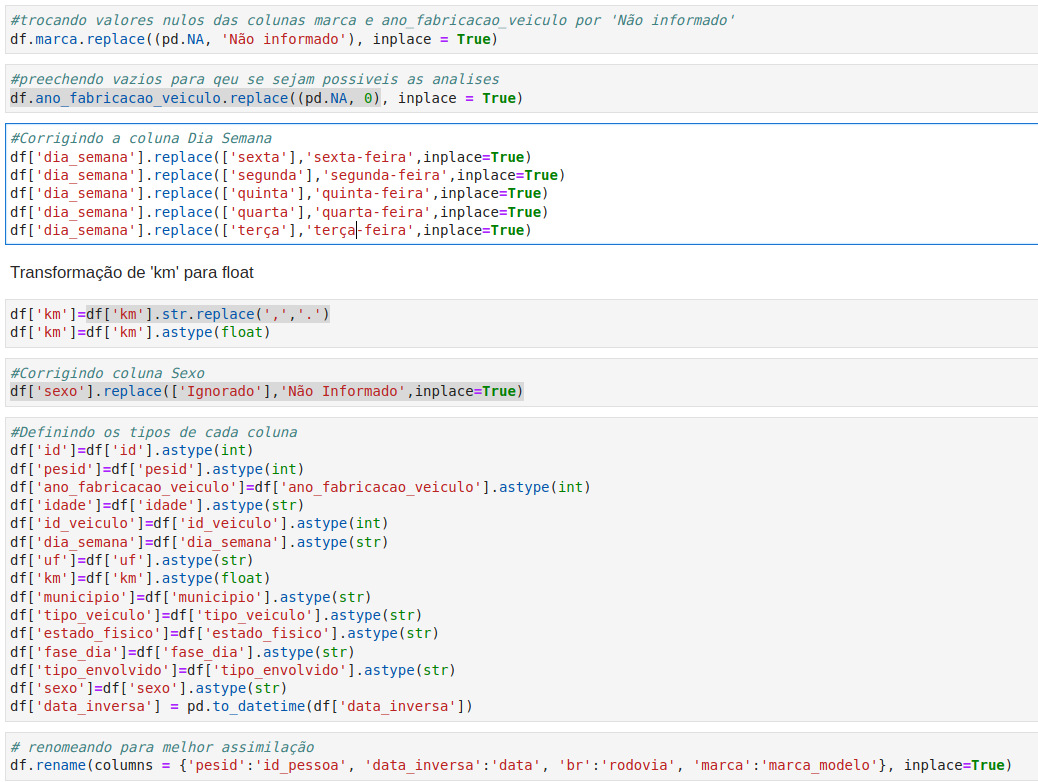
*Acidentes Agrupados por pessoas*

|  |  |
| --- | --- |
| **NOME DA VARIÁVEL** | **DESCRIÇAO** |
| *id* | Variável com valores numéricos, representando o  identificador do acidente. |
| *pesid* | Variável com valores numéricos, representando o  identificador da pessoa envolvida. |
| *data\_inversa* | Data da ocorrência no formato dd/mm/aaaa. |
| *dia\_semana* | Dia da semana da ocorrência. Ex.: Segunda, Terça, etc. |
| *horario* | Horário da ocorrência no formato hh:mm:ss. |
| *uf* | Unidade da Federação. Ex.: MG, PE, DF, etc. |
| *br* | Variável com valores numéricos, representando o  identificador da BR do acidente. |
| *km* | Identificação do quilômetro onde ocorreu o acidente, com valor mínimo de 0,1 km e com a casa decimal separada  por ponto. |
| *municipio* | Nome do município de ocorrência do acidente |
| *causa\_acidente* | Identificação da causa principal do acidente. Neste conjunto de dados são excluídos os acidentes com a  variável causa principal igual a “Não”. |
| *tipo\_acidente* | Identificação do tipo de acidente. Ex.: Colisão frontal, Saída de pista, etc. Neste conjunto de dados são excluídos os tipos de acidentes com ordem maior ou igual a dois. A ordem do acidente demonstra a sequência cronológica dos  tipos presentes na mesma ocorrência. |
| *classificação\_acidente* | Classificação quanto à gravidade do acidente: Sem  Vítimas, Com Vítimas Feridas, Com Vítimas Fatais e Ignorado. |
| *fase\_dia* | Fase do dia no momento do acidente. Ex. Amanhecer, Pleno dia, etc. |
| *sentido\_via* | Sentido da via considerando o ponto de colisão: Crescente e decrescente. |
| *condição\_meteorologica* | Condição meteorológica no momento do acidente: Céu  claro, chuva, vento, etc. |
| *tipo\_pista* | Tipo da pista considerando a quantidade de faixas: Dupla,  simples ou múltipla. |
| *tracado\_via* | Descrição do traçado da via. |
| *uso\_solo* | Descrição sobre as características do local do acidente:  Urbano=Sim;Rural=Não. |
| *id\_veiculo* | Variável com valores numéricos, representando o  identificador do veículo envolvido. |

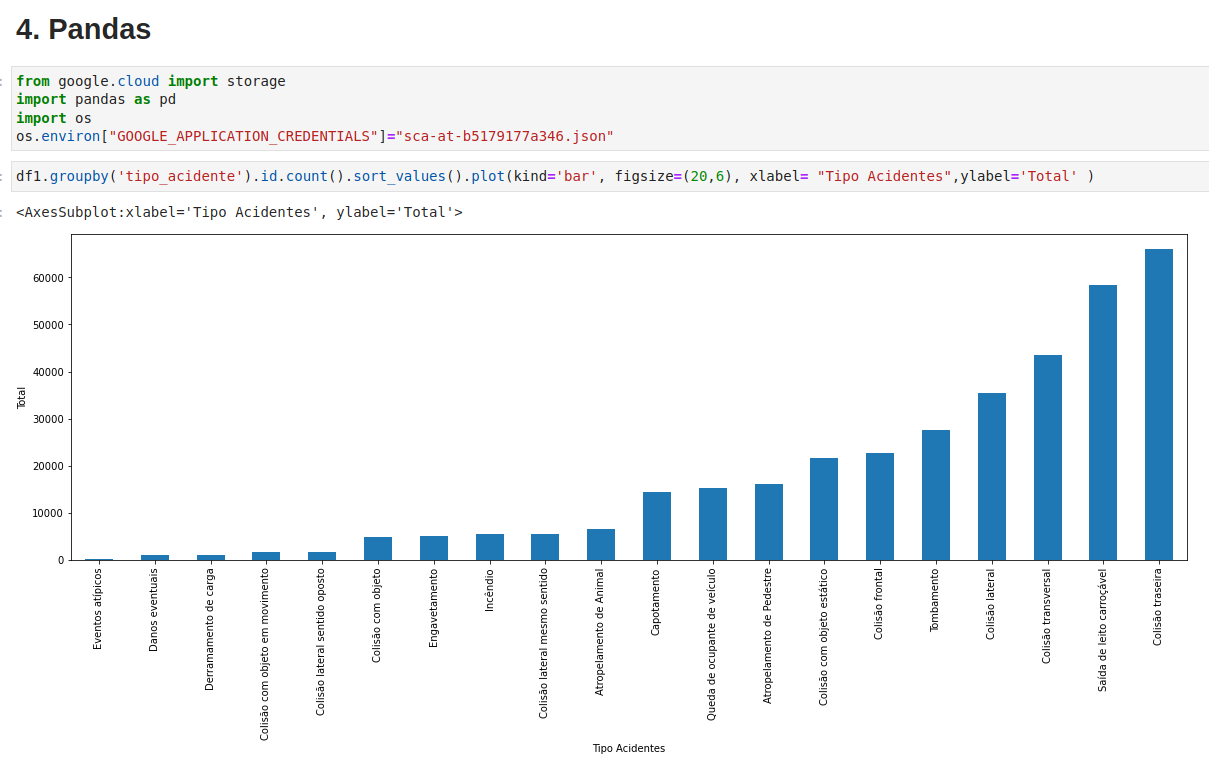
|  |  |
| --- | --- |
| *tipo\_veiculo* | Tipo do veículo conforme Art. 96 do Código de Trânsito  Brasileiro. Ex.: Automóvel, Caminhão, Motocicleta, etc. |
| *marca* | Descrição da marca do veículo. |
| *ano\_fabricacao\_veiculo* | Ano de fabricação do veículo, formato aaaa |
| *tipo\_envolvido* | Tipo de envolvido no acidente conforme sua participação  no evento. Ex.: condutor, passageiro, pedestre, etc |
| *estado\_fisico* | Condição do envolvido conforme a gravidade das lesões.  Ex.: morto, ferido leve, etc. |
| *idade* | Idade do envolvido. O código “-1” indica que não foi  possível coletar tal informação. |
| *sexo* | Sexo do envolvido. O valor “inválido” indica que não foi  possível coletar tal informação. |
| *ilesos* | Valor binário que identifica se o envolvido foi  classificado como ileso. |
| *feridos\_leves* | Valor binário que identifica se o envolvido foi  classificado como ferido leve. |
| *feridos\_graves* | Valor binário que identifica se o envolvido foi  classificado como ferido grave. |
| *mortos* | Valor binário que identifica se o envolvido foi  classificado como morto. |
| *latitude* | Latitude do local do acidente em formato geodésico  decimal. |
| *longitude* | Longitude do local do acidente em formato geodésico  decimal. |

**Pandas**

**Pandas**

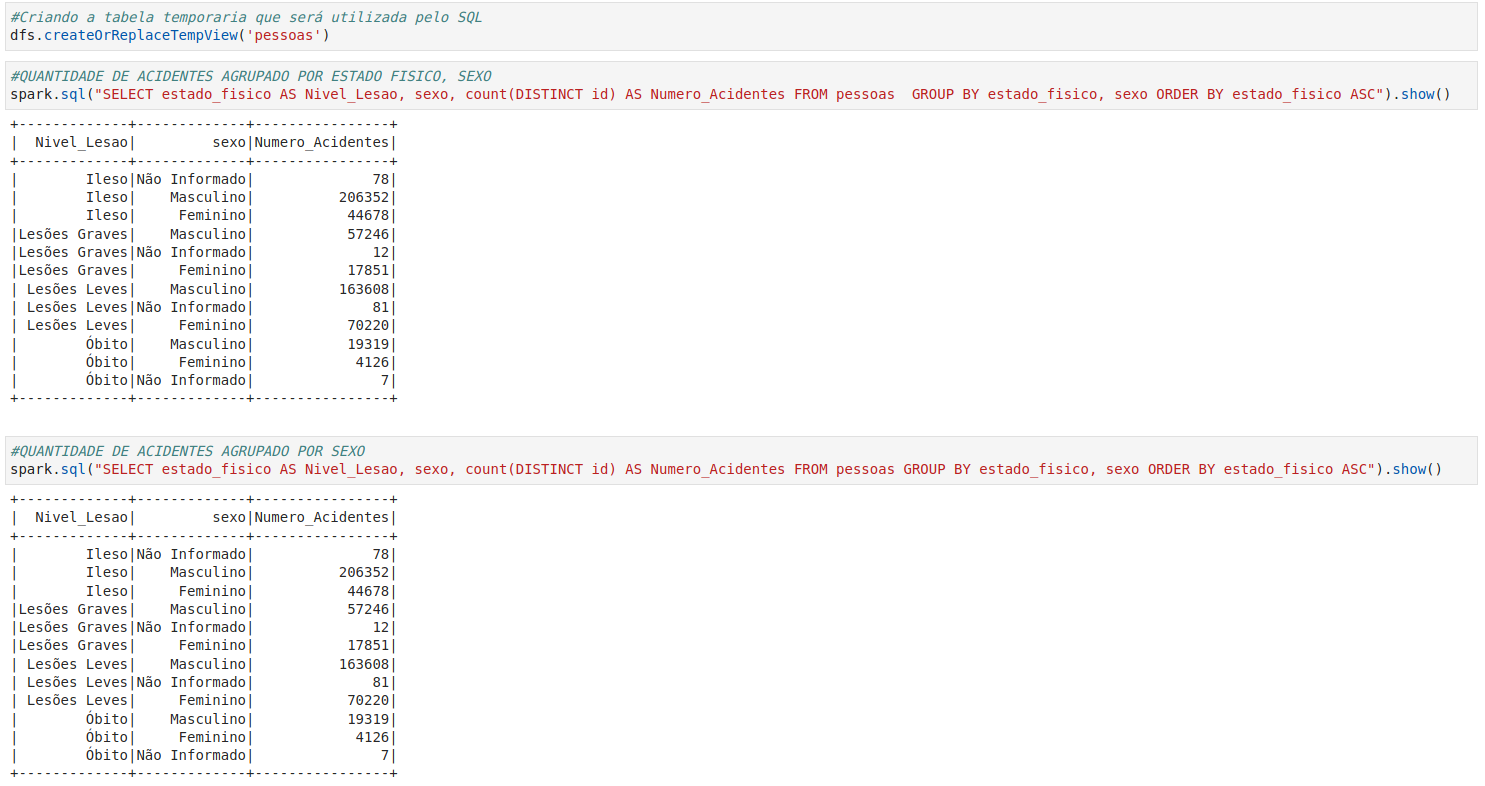


**Plotagens Pandas**



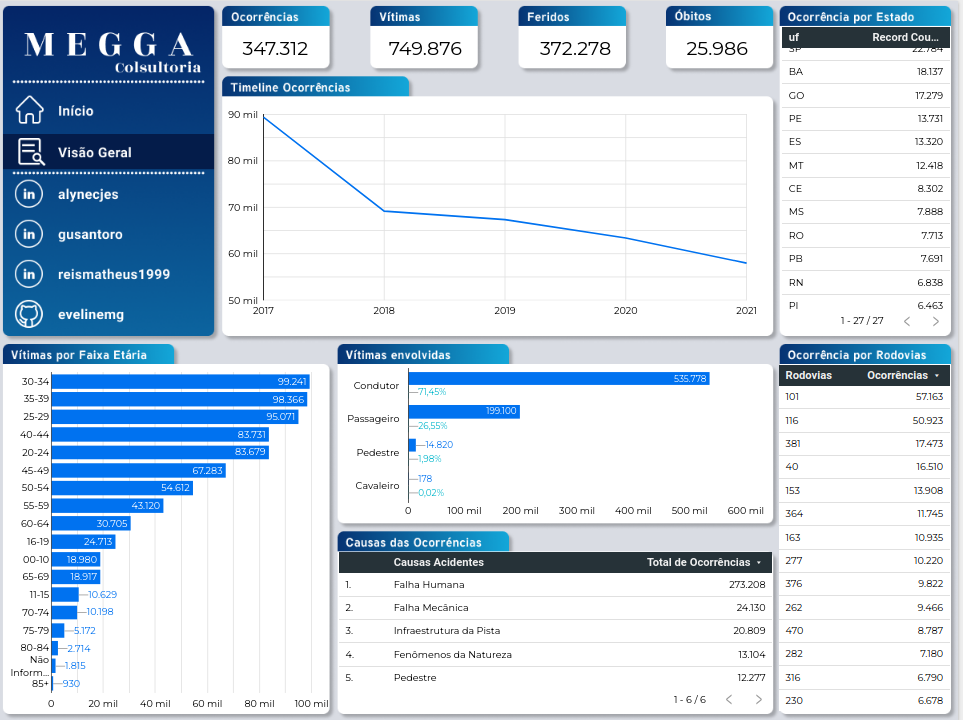
**PySpark**

**Spark SQL**

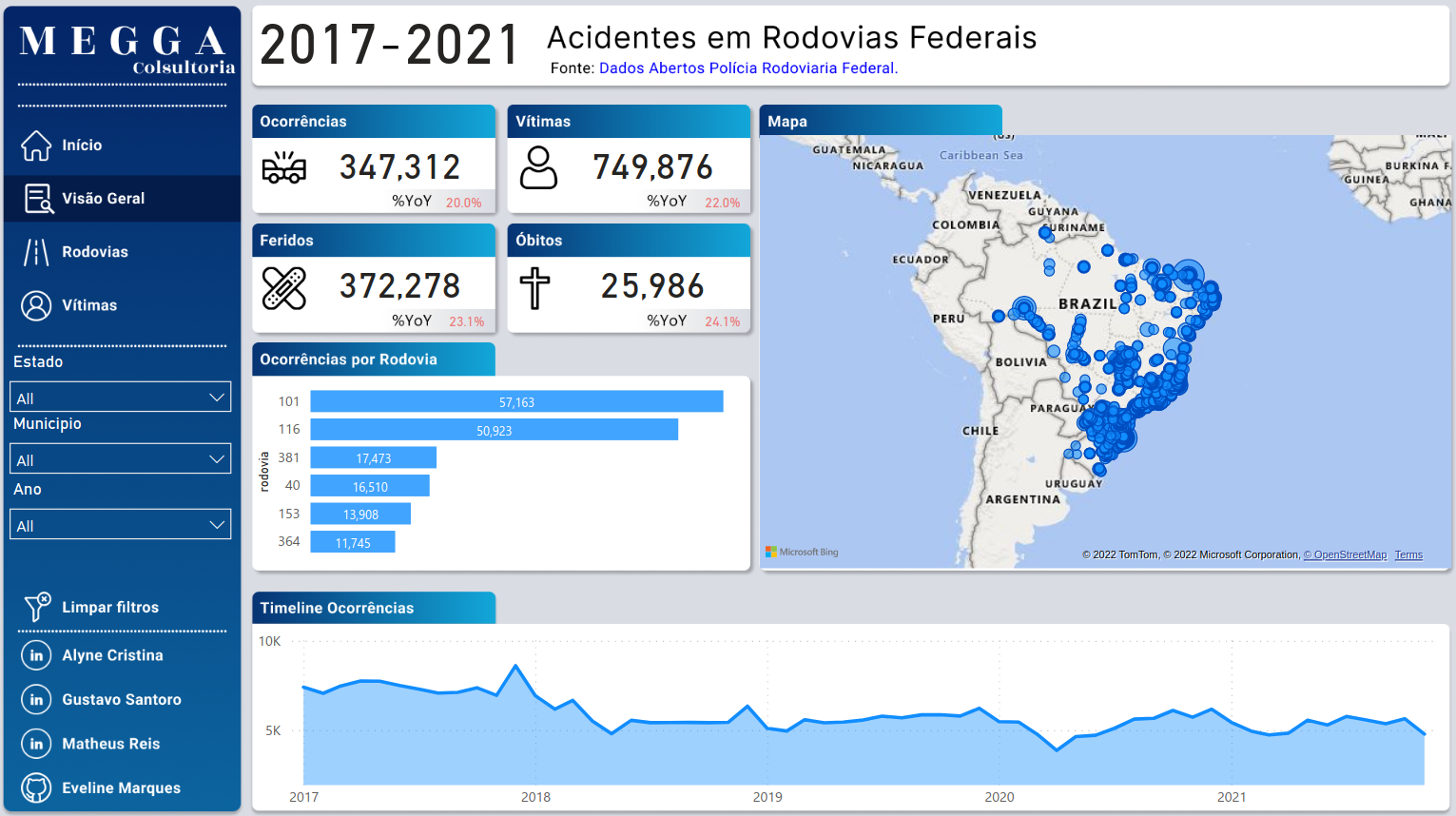
****

**Análises**

**DataStudio**

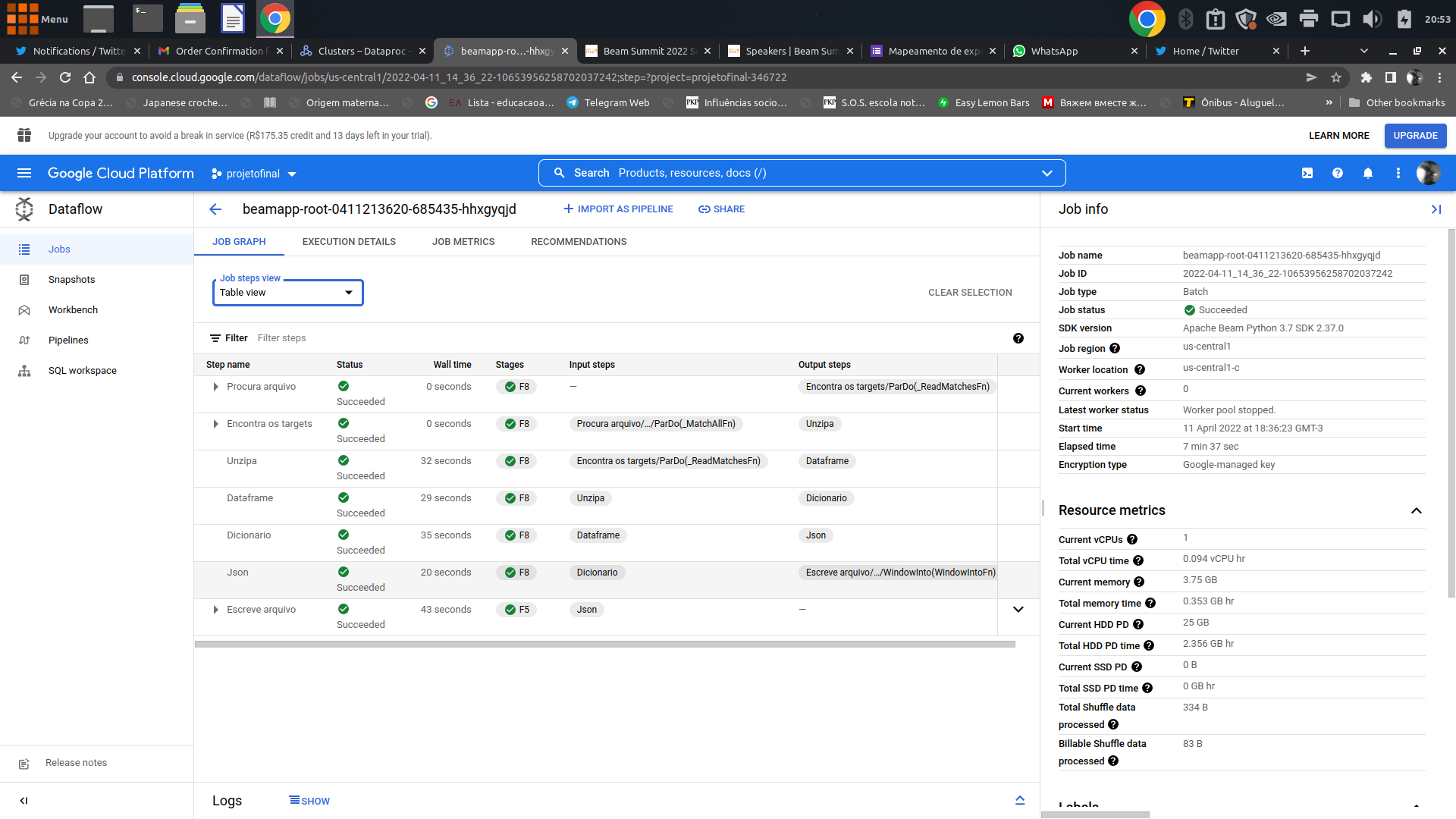
**[](https://datastudio.google.com/reporting/bf781d54-f1d6-4983-9447-6bccf32cecbf/page/p_07ah5vnltc)**

**PowerBI**

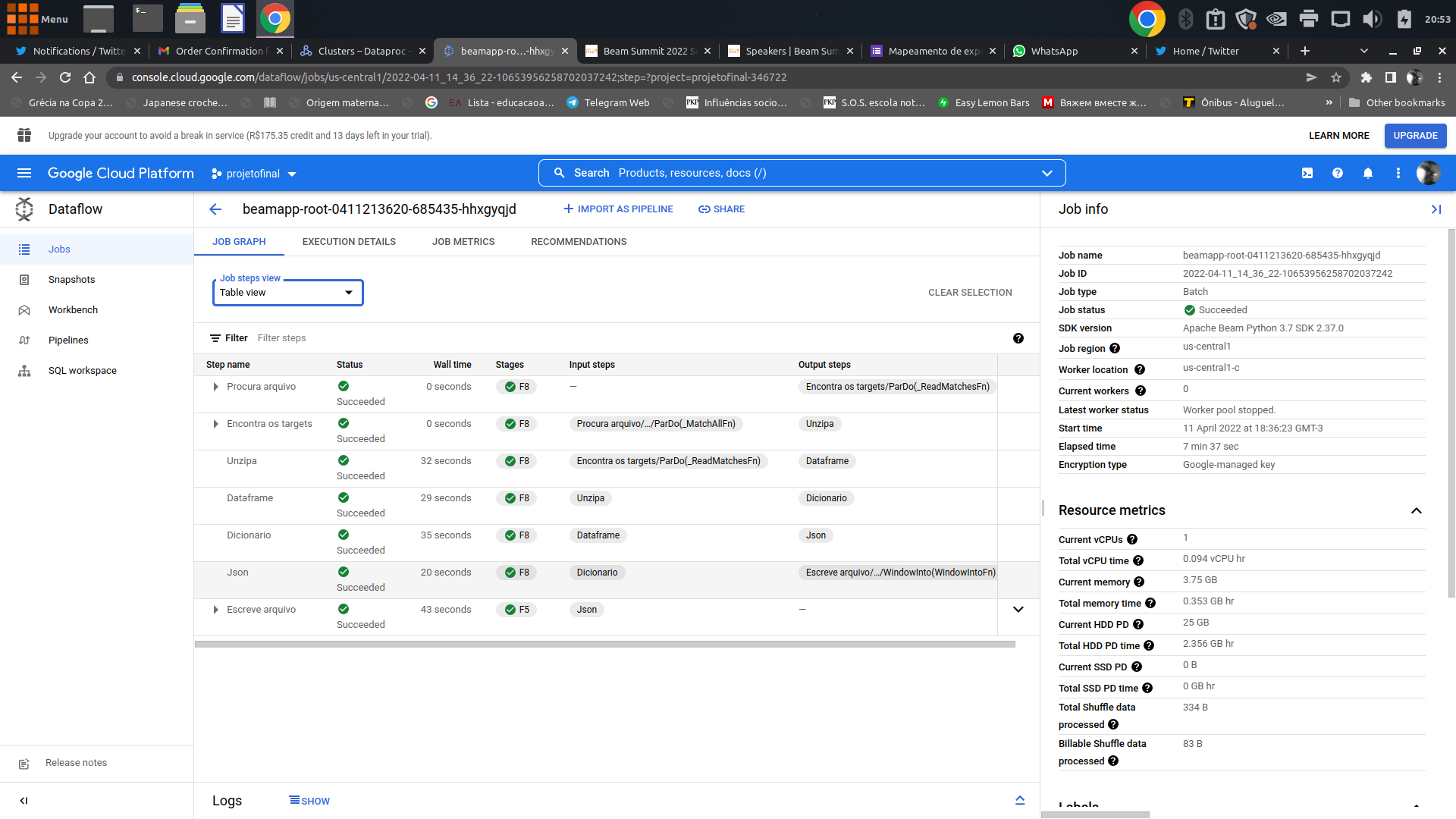
**[](https://app.powerbi.com/view?r=eyJrIjoiYTI1Y2ExZWUtOGUyYi00YjZhLTlhMTQtNzA4NzY0ZGU5ODg0IiwidCI6ImQwZmYyY2NkLTI5MTItNDFhMS05NmM4LTA3NTU3NTg1OTkyYiJ9)**

**Dataflow**

**Pipelines**

****

**Pipelines**

****

**Mapeamento dos tratamentos e normalizações nos data sets Pessoas e Ocorrências**

| Dataset Pessoas | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Database: | | Acidentes por pessoas | | | | | | colunas | 35 |
| Descrição | | Tabela de acidentes agrupados por cada indíviduo envolvido na ocorrência. | | | | | | linhas | 829727 |
| local: | | gs://projetofinalsc/dadosbrutos/pessoas/acidentes\_pessoas.json | | | | | | | |
|  | Origem | | | | Objetivo | | | | |
| **nº** | **campo** | | **data type** | **descrição** | **campo** | **data type** | **transformação** | | |
| 1 | id | | inteiro | PK ocorrência | id | int |  | | |
| 2 | pesid | | float | PK pessoa | id\_pessoa | int |  | | |
| 3 | data\_invertida | | string | data YYYY-MM-DD | data | datetime |  | | |
| 4 | dia\_semana | | string |  |  | string | df['dia\_semana'].replace(['{dia}'],'{dia}-feira',inplace=True) | | |
| 5 | horario | | string |  |  |  | remove | | |
| 6 | uf | | string |  |  | string |  | | |
| 7 | br | | string |  | rodovia | string |  | | |
| 8 | km | | float |  |  | float | df['km'].str.replace(',','.') | | |
| 9 | municipio | | string |  |  | string |  | | |
| 10 | causa\_acidente | | string |  |  |  | remove | | |
| 11 | tipo\_acidente | | string |  |  |  | remove | | |
| 12 | classificacao\_acidente | | string |  |  |  | remove | | |
| 13 | fase\_dia | | string |  |  |  | remove | | |
| 14 | sentido\_via | | string |  |  |  | remove | | |
| 15 | condicao\_metereologica | | string |  |  |  | remove | | |
| 16 | tipo\_pista | | string |  |  |  | remove | | |
| 17 | tracado\_via | | string |  |  |  | remove | | |
| 18 | uso\_solo | | string | Urbano = sim; Rural=não |  | string |  | | |
| 19 | id\_veiculo | | float |  |  | int |  | | |
| 20 | tipo\_veiculo | | string |  |  | string |  | | |
| 21 | marca | | string |  | marca\_modelo | string |  | | |
| 22 | ano\_fabricacao\_veiculo | | float |  |  | int | Nível Pandas  df.ano\_fabricacao\_veiculo.replace((pd.NA, 0) | | |
| 23 | tipo\_envolvido | | string |  |  | string |  | | |
| 24 | estado\_fisico | | string |  |  | string |  | | |
| 25 | idade | | float | -1 idade não foi coletada | faixa\_idade | string | Nível PySpark  dfs.withColumn('faixa\_idade', F.when((col('idade')>100), lit('Não Informado'))  .when((col('idade')>85), lit('85+'))  .when((col('idade')>=80), lit('80-84'))  .when((col('idade')>=75), lit('75-79'))  .when((col('idade')>=70), lit('70-74'))  .when((col('idade')>=65), lit('65-69'))  .when((col('idade')>=60), lit('60-64'))  .when((col('idade')>=55), lit('55-59'))  .when((col('idade')>=50), lit('50-54'))  .when((col('idade')>=45), lit('45-49'))  .when((col('idade')>=40), lit('40-44'))  .when((col('idade')>=35), lit('35-39'))  .when((col('idade')>=30), lit('30-34'))  .when((col('idade')>=25), lit('25-29'))  .when((col('idade')>=20), lit('20-24'))  .when((col('idade')>=16), lit('16-19'))  .when((col('idade')>=11), lit('11-15'))  .when((col('idade')<=10), lit('00-10'))) | | |
| 26 | sexo | | string |  |  | string | Nível Pandas  df['sexo'].replace(['Ignorado'],'Não Informado',inplace=True) | | |
| 27 | ilesos | | int | 0- não; 1-sim booleano |  |  | remove | | |
| 28 | feridos\_leves | | int | 0- não; 1-sim booleano |  |  | remove | | |
| 29 | feridos\_graves | | int | 0- não; 1-sim booleano |  |  | remove | | |
| 30 | mortos | | int | 0- não; 1-sim booleano |  |  | remove | | |
| 31 | latitude | | string |  |  | float | Nível Pandas  df['latitude'].str.replace(',','.') | | |
| 32 | longitude | |  |  |  | float | Nível Pandas  df['longitude'].str.replace(',','.') | | |
| 33 | regional | | string |  |  |  | remove | | |
| 34 | delegacia | | string |  |  |  | remove | | |
| 35 | uop | | string |  |  |  | remove | | |

| Dataset Ocorrências | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Database: | | Acidentes por ocorrências | | | | | | colunas | 30 |
| Descrição | | Tabela de acidentes agrupados por cada ocorrência | | | | | | linhas | 354293 |
| Local: | | gs://projetofinalsc/dadosbrutos/pessoas/acidentes\_ocorrencia.csv | | | | | | | |
|  | Origem | | | | Objetivo | | | | |
| **nº** | **campo** | | **data type** | **descrição** | **campo** | **data type** | **transformação** | | |
| 1 | id | | inteiro | PK ocorrência | id | int |  | | |
| 2 | data\_inversa | | string | data YYYY-MM-DD | data | datetime |  | | |
| 3 | dia\_semana | | string |  |  | string | Nível  df['dia\_semana'].replace(['{dia}'],'{dia}-feira',inplace=True) | | |
| 4 | horario | | string |  |  |  | remove | | |
| 5 | uf | | string |  |  | string |  | | |
| 6 | br | | string |  | rodovia | string |  | | |
| 7 | km | | float |  |  | float | Nível Pandas  df['km'].str.replace(',','.') | | |
| 8 | municipio | | string |  |  | string |  | | |
| 9 | causa\_acidente | | string |  |  | string |  | | |
| 10 | tipo\_acidente | | string |  |  | string |  | | |
| 11 | classificacao\_acidente | | string |  |  | string |  | | |
| 12 | fase\_dia | | string |  |  | string |  | | |
| 13 | sentido\_via | | string |  |  | string |  | | |
| 14 | condicao\_metereologica | | string |  |  | string |  | | |
| 15 | tipo\_pista | | string |  |  | string |  | | |
| 16 | tracado\_via | | string |  |  | string |  | | |
| 17 | uso\_solo | | string | Urbano = sim; Rural=não |  | string |  | | |
| 18 | pessoas | | int |  |  | int |  | | |
| 19 | veiculos | | string |  |  | string |  | | |
| 20 | ilesos | | int | 0- não; 1-sim booleano |  | int |  | | |
| 21 | feridos\_leves | | int | 0- não; 1-sim booleano |  | int |  | | |
| 22 | feridos\_graves | | int | 0- não; 1-sim booleano |  | int |  | | |
| 23 | ignorados | | int |  |  | int |  | | |
| 24 | mortos | | int | 0- não; 1-sim booleano |  | int |  | | |
| 25 | latitude | | string |  |  | float | Nível Pandas  df['latitude'].str.replace(',','.') | | |
| 26 | longitude | |  |  |  | float | Nível Pandas  df['longitude'].str.replace(',','.') | | |
| 26 | regional | | string |  |  |  | remove | | |
| 27 | delegacia | | string |  |  |  | remove | | |
| 28 | uop | | string |  |  |  | remove | | |
| 29 |  | |  | A partir da coluna causa\_acidente | consumo\_alcool | string | Nível PySpark:  df=df.withColumn("consumo\_alcool", F.col('causa\_acidente').rlike("Álcool"))  df=df.withColumn('consumo\_alcool', when(df.consumo\_alcool=='true', 'sim').otherwise('não')) | | |
| 30 |  | |  | A partir de da coluna causa\_acidente | grupo\_causas | string | Nível PySpark  b=[x[0] for x in df.select('causa\_acidente').distinct().collect()]  df=df.withColumn("causa\_grupos", condicao) | | |

**Conclusão**

Com os passos descritos nesse documento é possível repetir as operações de tratamento nos datasets utilizados de acordo com a necessidade do trabalho. Documentar cada transformação necessária nos diversos campos e a geração dos dados finais colabora para a organização e manutenção do trabalho.

Durante a elaboração desse trabalho também foi possível notar que as decisões tomadas em uma planilha impactaria a outra, sendo assim necessário, fazer remoções e alterações compatíveis em ambos datasets.

Além disso, automatizar processos como a pipeline pode impactar na organização e redução de tempo demandado em certas tarefas, como descomprimir e concatenar todos os arquivos.

Ao final, foi possível fazer várias análises sobre os dados trabalhados e a construção dos relatórios em ferramentas diferentes trouxe a possibilidade de visualização dos dados em diversas perspectivas, com filtros trazendo diversos níveis de analises, por estado, rodovia, por ano, entre outros.