NKVS Consultoria

SOUL CODE

PROJETO FINAL

ENGENHARIA DE DADOS - ED7

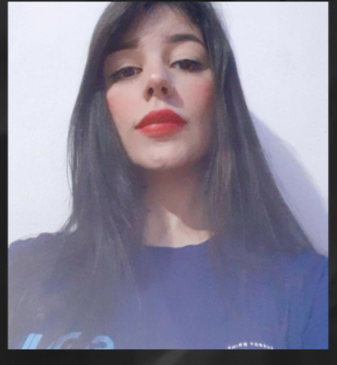
TEMA: MERCADO AUTOMOTIVO

NKVS
Consultoria

NOSSA EQUIPE

NAIARA GAMA

KARINY ALPIANO



KELI ALVES

VICTOR SILVA

SANDRA GONÇALVES

Requisitos necessários para a elaboração do Projeto

- Uso de no mínimo dois datasets em formatos diferentes, sendo que um deles tem que ser obrigatoriamente em formato CSV.
- Procedimentos de ETL (extração, transformação, carregamento) e análises feitas por meio de Pandas e Pyspark.
- Os datasets devem ser salvos e operados em armazenamento Cloud dentro da plataforma GCP.
- O armazenamento dos dados originais deve ser feito em MySQL e dos dados tratados deve ser em datalake (Gstorage) , DW (BigQuery) ou em ambos.
- Deve ser feita análises dentro do Big Query utilizando a linguagem padrão SQL com a descrição das consultas feitas.
- Deve ser criado no Looker Studio um dashboard para exibição gráfica dos dados tratados trazendo insights importantes.
- E deve ser demonstrado em um workflow simples (gráfico) as etapas de ETL com suas respectivas ferramentas.

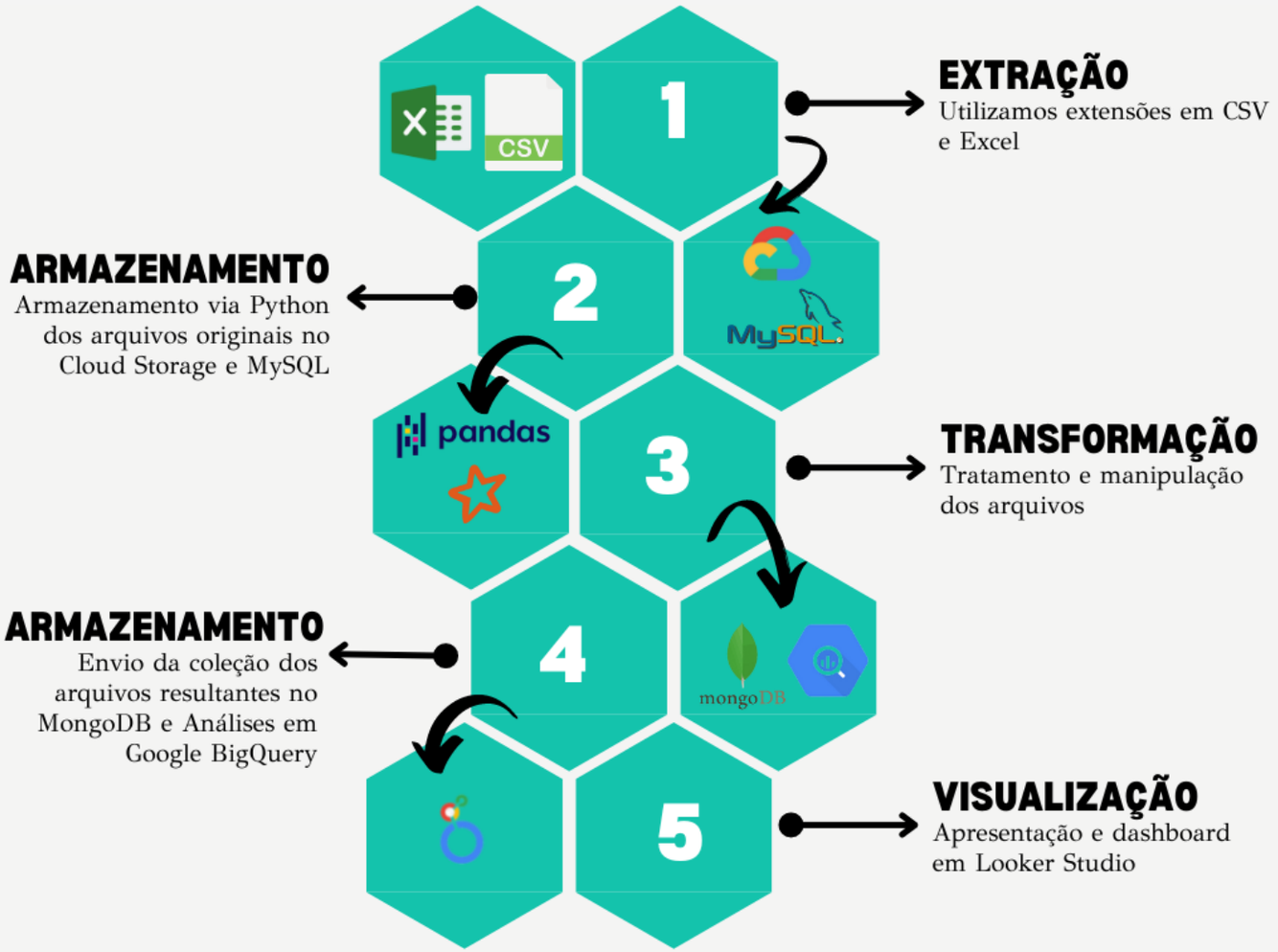
Projeto

- Este projeto tem o objetivo de realizar um processo de ETL para analisar o mercado automotivo Brasileiro.

- Com base no escopo, analisamos dois datasets essenciais para o projeto, sendo eles as tabelas FIPE 2022 e ANFAVEA Brasil.

- Nossa amostragem será demonstrada no período de 2018 a 2022 (antes e durante a pandemia do Covid-19), focando em dados sobre produção, vendas e exportação de automóveis, e também, bases de preços e características dos veículos comercializados no Brasil.

Workflow


1
EXTRAÇÃO
Utilizamos extensões em CSV e Excel
CSV
2
ARMAZENAMENTO
Armazenamento via Python dos arquivos originais no Cloud Storage e MySQL
MySQL
pandas
3
TRANSFORMAÇÃO
Tratamento e manipulação dos arquivos
4
ARMAZENAMENTO
Envio da coleção dos arquivos resultantes no MongoDB e Análises em Google BigQuery
mongoDB
5
VISUALIZAÇÃO
Apresentação e dashboard em Looker Studio

Tratamento com Pandas

Tabela FIPE 2022 - Renomeação das colunas

```
[ ]
#Tradução do nome das colunas
df2.rename(columns={'year_of_reference':'ano_referencia',
                    'month_of_reference':'mes_referencia',
                    'fipe_code':'codigo_fipe',
                    'authentication':'autenticacao',
                    'brand':'marca',
                    'model':'modelo',
                    'fuel':'combustivel',
                    'gear':'cambio',
                    'engine_size':'tamanho_motor',
                    'year_model':'ano_modelo',
                    'avg_price_brl':'preco_medio_reais',
                    'age_years':'anos_idade'
                   }, inplace=True)
```

Tratamento com Pandas

Tabela FIPE 2022 - Renomeação e tradução das linhas

Traduzindo e renomeando Linhas das Colunas

```
[ ]  1 df2["combustivel"] = df2["combustivel"].replace("Gasoline", "Gasolina")
```

```
[ ]  1 df2["combustivel"] = df2["combustivel"].replace("Alcohol", "Alcool")
```

```
[ ]  1 df2["mes_referencia"] = df2["mes_referencia"].replace("January", "Janeiro")
```

```
[ ]  1 df2["mes_referencia"] = df2["mes_referencia"].replace("February", "Fevereiro")
```

```
[ ]  1 df2["mes_referencia"] = df2["mes_referencia"].replace("March", "Março")
```

```
[ ]  1 df2["mes_referencia"] = df2["mes_referencia"].replace("April", "Abril")
```

```
[ ]  1 df2["mes_referencia"] = df2["mes_referencia"].replace("May", "Maio")
```

```
[ ]  1 df2["mes_referencia"] = df2["mes_referencia"].replace("June", "Junho")
```

```
[ ]  1 df2["mes_referencia"] = df2["mes_referencia"].replace("July", "Julho")
```

```
[ ]  1 df2["mes_referencia"] = df2["mes_referencia"].replace("August", "Agosto")
```

```
[ ]  1 df2["mes_referencia"] = df2["mes_referencia"].replace("October", "Outubro")
```

```
[ ]  1 df2["mes_referencia"] = df2["mes_referencia"].replace("September", "Setembro")
```

```
[ ]  1 df2["mes_referencia"] = df2["mes_referencia"].replace("November", "Novembro")
```

```
[ ]  1 df2["mes_referencia"] = df2["mes_referencia"].replace("December", "Dezembro")
```

```
[ ]  1 df2['cambio'] = df2['cambio'].replace('automatic', 'automatico')
```

Tratamento com PySpark

Criando SparkSession

```
[ ]  1 #Configurando a variável do ambiente da sessão spark
     2 spark = (
     3     SparkSession.builder
     4                 .master('local')
     5                 .appName('dataset')
     6                 .config('spark.ui.port','4050')
     7                 .getOrCreate()
     8 )
```

```
[ ]  1 spark
```

SparkSession - in-memory

SparkContext

Spark UI

Version
v3.3.1
Master
local
AppName
dataset

Tratamento com PySpark

```
# Criando dataframe
df = (
    spark.createDataFrame(pd.read_excel('/content/dados_brasil.xlsx'))

)
#Passamos para o Pandas para tratar as colunas tipo float, arredondando para
#duas casas decimais
df = df.toPandas()
df = df.round(2)
#Visualizando o df
df1 = spark.createDataFrame(df)
df1.show()
```

```
+-------------------+------+--------------+--------------------------+-------------------------------------+-
|               Date|Entity|Bus production|Bus production - cumsum_12|Bus production - current_prices_3m_yoy|E
+-------------------+------+--------------+--------------------------+-------------------------------------+-
|1981-01-01 00:00:00|Brazil|           NaN|                       NaN|                                  NaN|
|1981-02-01 00:00:00|Brazil|           NaN|                       NaN|                                  NaN|
|1981-03-01 00:00:00|Brazil|           NaN|                       NaN|                                  NaN|
|1981-04-01 00:00:00|Brazil|           NaN|                       NaN|                                  NaN|
|1981-05-01 00:00:00|Brazil|           NaN|                       NaN|                                  NaN|
|1981-06-01 00:00:00|Brazil|           NaN|                       NaN|                                  NaN|
|1981-07-01 00:00:00|Brazil|           NaN|                       NaN|                                  NaN|
|1981-08-01 00:00:00|Brazil|           NaN|                       NaN|                                  NaN|
|1981-09-01 00:00:00|Brazil|           NaN|                       NaN|                                  NaN|
|1981-10-01 00:00:00|Brazil|           NaN|                       NaN|                                  NaN|
|1981-11-01 00:00:00|Brazil|           NaN|                       NaN|                                  NaN|
|1981-12-01 00:00:00|Brazil|           NaN|                       NaN|                                  NaN|
```

Drop Colunas

```
#Drop de colunas que não serão utilizadas
df1 = df1.drop('Bus production',
           'Bus production - cumsum_12',
           'Bus production - current_prices_3m_yoy',
           'Bus production - current_prices_mom',
           'Bus production - current_prices_yoy',
           'Bus production - sa',
           'Bus production - sa_MoM',
           'Four wheel tractor production',
           'Four wheel tractor production - cumsum_12',
           'Four wheel tractor production - current_prices_3m_yoy',
           'Four wheel tractor production - current_prices_mom',
           'Four wheel tractor production - current_prices_yoy',
           'Four wheel tractor production - sa',
           'Four wheel tractor production - sa_MoM',
           'Other agricultural machinery',
           'Other agricultural machinery - cumsum_12',
           'Other agricultural machinery - current_prices_3m_yoy',
           'Other agricultural machinery - current_prices_mom',
           'Other agricultural machinery - current_prices_mom',
           'Other agricultural machinery - current_prices_yoy',
           'Other agricultural machinery - sa',
           'Other agricultural machinery - sa_MoM',
           'Production of agricultural machinery (total)',
           'Production of agricultural machinery (total) - cumsum_12',
           'Production of agricultural machinery (total) - current_prices_3m_yoy',
           'Production of agricultural machinery (total) - current_prices_mom',
```

Schema

Criando Schema

```
[ ]
#Esquema
esquema = (
    StructType([
        StructField('Date', DateType()),
        StructField('Domestic vehicle sales - Seasonally Adjusted', FloatType()),
        StructField('Domestic vehicle sales - Seasonally Adjusted - current_prices_mom', FloatType()),
        StructField('Domestic vehicle sales - Seasonally Adjusted - current_prices_yoy', FloatType()),
        StructField('Passenger cars and light commercial vehicles production', FloatType()),
        StructField('Passenger cars and light commercial vehicles production - current_prices_mom', FloatType()),
        StructField('Passenger cars and light commercial vehicles production - current_prices_yoy', FloatType()),
        StructField('Vehicle exports - Seasonally Adjusted', FloatType()),
        StructField('Vehicle exports - Seasonally Adjusted - current_prices_mom', FloatType()),
        StructField('Vehicle exports - Seasonally Adjusted - current_prices_yoy', FloatType()),
        StructField('Vehicle exports - Seasonally Adjusted - sa', FloatType()),
        StructField('Vehicle exports - Seasonally Adjusted - sa_MoM', FloatType()),
        StructField('Vehicle sales (total) - Seasonally Adjusted', FloatType()),
        StructField('Vehicle sales (total) - Seasonally Adjusted - current_prices_mom', FloatType()),
        StructField('Vehicle sales (total) - Seasonally Adjusted - current_prices_yoy', FloatType()),
        StructField('Vehicles production (total)', FloatType()),
        StructField('Vehicles production (total) - current_prices_mom', FloatType()),
        StructField('Vehicles production (total) - current_prices_yoy', FloatType()),
        StructField('Vehicles production (total) - sa', FloatType()),
        StructField('Vehicles production (total) - sa_MoM', FloatType()),
```

Alteração do Formato do Arquivo

```
[ ]  1 #Alteração para csv
     2 df1.write.format('csv').mode('overwrite').save('/content/projeto_final_brasil.csv')
```

```
[ ]  1 df2 = (
     2     spark.read.format('csv')
     3                .option('header', 'true')
     4                .option('inferschema', 'true')
     5                .option('delimiter', ',')
     6                .load('/content/projeto_final_brasil.csv', schema = esquema)
     7 )
```

Tradução e Renomeação

Renomear e Traduzir as colunas

```
[ ]  1 #Colunas traduzidas e renomeadas
     2 df3 = ( df2.withColumnRenamed('Date','data')
     3            .withColumnRenamed('Domestic vehicle sales - Seasonally Adjusted','vendas_saz')
     4            .withColumnRenamed('Domestic vehicle sales - Seasonally Adjusted - current_prices_mom','vendas_saz_preco_mes')
     5            .withColumnRenamed('Domestic vehicle sales - Seasonally Adjusted - current_prices_yoy','vendas_saz_preco_ano')
     6            .withColumnRenamed('Passenger cars and light commercial vehicles production','prod_auto_leve')
     7            .withColumnRenamed('Passenger cars and light commercial vehicles production - current_prices_mom','prod_auto_leve_preco_atual_mes')
     8            .withColumnRenamed('Passenger cars and light commercial vehicles production - current_prices_yoy','prod_auto-leve_ano')
     9            .withColumnRenamed('Vehicle exports - Seasonally Adjusted','expo_saz')
    10            .withColumnRenamed('Vehicle exports - Seasonally Adjusted - current_prices_mom','expo_saz_preço_mes')
    11            .withColumnRenamed('Vehicle exports - Seasonally Adjusted - current_prices_yoy','expo__saz_preco_ano')
    12            .withColumnRenamed('Vehicle exports - Seasonally Adjusted - sa','expo_saz_sa')
    13            .withColumnRenamed('Vehicle exports - Seasonally Adjusted - sa_MoM ','expo_saz_sa_mes')
    14            .withColumnRenamed('Vehicle sales (total) - Seasonally Adjusted','vendas_total_saz')
    15            .withColumnRenamed('Vehicle sales (total) - Seasonally Adjusted - current_prices_mom','vendas_total_saz_preco_mes')
    16            .withColumnRenamed('Vehicle exports - Seasonally Adjusted - sa_MoM ','expo_sa_mes')
    17            .withColumnRenamed('Vehicle sales (total) - Seasonally Adjusted - current_prices_yoy','vendas_total_saz_preco_ano')
    18            .withColumnRenamed('Vehicles production (total)','prod_total')
    19            .withColumnRenamed('Vehicles production (total) - current_prices_mom','prod_total_preco_mes')
    20            .withColumnRenamed('Vehicles production (total) - current_prices_yoy','prod_total_preco_ano')
    21            .withColumnRenamed('Vehicles production (total) - sa','prod_total_sa')
    22            .withColumnRenamed('Vehicles production (total) - sa_MoM','prod_total_sa_mes')
    23            .withColumnRenamed('Vehicles production (total) - Seasonally Adjusted','prod_total_saz')
    24            .withColumnRenamed('Vehicle sales (total) - Seasonally Adjusted ','vendas_total_saz')
```

Criação de Coluna

```
[ ]  1 #Separação por '-'
     2 split_cols = pyspark.sql.functions.split(df4['data'], '-')
```

```
[ ]  1 #Criando uma nova coluna 'ano'
     2 df5 = df4.withColumn('ano', split_cols.getItem(0))
     3 df5.show()
```

```
---+---------------+---------------+---------+-------------+---------+-------------+-------------+-----------+--------------------+------------------+----+
las|vendas_preco_mes|vendas_preco_ano|vendas_sa|vendas_sa_mes|expo_auto|expo_preco_mes|expo_preco_ano|vendas_total|vendas_total_preco_mes|vendas_total_sa_mes| ano|
---+---------------+---------------+---------+-------------+---------+-------------+-------------+-----------+--------------------+------------------+----+
.0|          -6.08|          34.07| 62062.05|        -1.11|  11215.0|        -0.15|         -19.3|    74537.0|               -5.24|              1.15|1982|
.0|         -15.63|          -0.79|  95027.3|        -7.53|   9046.0|         -24.5|         9.13|   106604.0|              -16.46|             -8.41|1994|
.0|         -18.64|           7.74| 115928.2|        -9.41|  37587.0|        -19.41|         34.62|   148432.0|              -18.83|             -11.6|2004|
.0|          -8.86|           5.02|207712.89|        -6.98|  35373.0|         39.02|        -32.83|   255989.0|                -4.3|             -1.47|2009|
.0|           8.18|            8.5|133239.77|        10.19|  50328.0|         22.61|         84.33|   160239.0|               12.33|              9.48|2005|
.0|          -9.92|          -14.1|200352.95|       -12.13|  53017.0|         16.45|         3.67|   262554.0|                -5.6|             -9.0|2011|
.0|           -1.0|          -9.71|133875.44|        -8.56|  29444.0|         23.89|         5.55|   178900.0|                2.38|             -5.31|2021|
.0|          78.18|          13.67| 63081.76|        24.72|  24457.0|         24.32|        -3.46|    82921.0|               57.99|             25.71|1990|
```

Filtros

```
[ ]  1 #Substituição de todos os NaN por 0
     2 df7 = df5.fillna(0)
```

```
[ ]  1 #Convertendo a coluna ano para inteiro
     2 df_year = df7.withColumn('ano', col("ano").cast("integer"))
     3 #Filtrando os anos entre 2018 e 2022
     4 df_consolidado = df_year.filter((col('ano') >= 2018) & (col('ano') <= 2022))
```

```
[ ]  1 #Para visualizar colunas em duplicidade
     2 for i in df_consolidado:
     3   print(i)
```

```
[ ]  1 df_consolidado.show()
```

```
+----------+----------+-------------------+-------------------+-------------+-----------------------------+-----------------+--------+-----------------+----
|      data|vendas_saz|vendas_saz_preco_mes|vendas_saz_preco_ano|prod_auto_leve|prod_auto_leve_preco_atual_mes|prod_auto-leve_ano|expo_saz|expo_saz_preço_mes|expo
+----------+----------+-------------------+-------------------+-------------+-----------------------------+-----------------+--------+-----------------+----
|2021-08-01|  141804.0|              -4.51|             -13.98|     148857.0|                          0.7|           -26.15| 28721.0|            23.26|
|2018-02-01|  175386.0|              -0.52|               8.93|     203610.0|                         -3.3|             4.37| 62857.0|             2.02|
|2018-10-01|  210175.0|               6.3|              18.64|     250018.0|                        18.09|             2.41| 41340.0|            -4.34|
|2022-09-01|  162401.0|              -2.98|              28.17|     189007.0|                       -12.97|            18.82| 31770.0|            -31.2|
|2018-06-01|  181826.0|               8.65|               4.62|     244732.0|                        20.51|            18.91| 60070.0|            10.73|
|2020-12-01|  190310.0|              -1.48|             -10.02|     197802.0|                        -12.1|             21.1| 40656.0|            -8.33|
|2021-01-01|  190624.0|               0.16|               0.45|     190156.0|                        -3.87|             3.83| 36744.0|            -9.62|
|2021-06-01|  160445.0|              -4.65|              36.76|     151286.0|                       -14.66|            65.36| 29905.0|            -6.44|
|2018-04-01|  201206.0|              10.42|              23.75|     254154.0|                        -0.24|            38.01| 72381.0|            13.69|
```

Envio dos Dados Tratados para GCP

```
[ ]  1 # fazendo o load para enviar o arquivo ao gcs
     2 path = '/content/dados_brasil_consolidado.csv'
     3
     4 df_consolidado.write.mode('overwrite').option('header',True).csv(path)
```

```
[ ]  1 #Função para fazer upload de arquivo no bucket
     2 def upload_blob(bucket, arquivo, destino):
     3     client = storage.Client()
     4     bucket = client.bucket(bucket)
     5     blob = bucket.blob(destino)
     6
     7     blob.upload_from_filename(arquivo)
     8     header='True'
     9
    10     print(
    11         f"Arquivo {arquivo} upado em {destino}."
    12     )
```

```
[ ]  1 #Upload do arquivo
     2 bucket = 'bc26_projeto_final_auto'
     3 arquivo = '/content/dados_brasil_consolidado.csv/part-00000-91ab60fc-1639-4ebd-a3a9-6166c83756ff-c000.csv'
     4 destino = 'tratados/dados_brasil_consolidado.csv'
     5 upload_blob(bucket, arquivo, destino)
```

```
Arquivo /content/dados_brasil_consolidado.csv/part-00000-91ab60fc-1639-4ebd-a3a9-6166c83756ff-c000.csv upado em tratados/dados_brasil_consolidado.csv.
```

Armazenamento dos dataframes resultantes na coleção do MongoDB

Conexão com o MongoDB e envio da coleção tabela FIPE 2022

MongoDB

```
[ ] #conector do mongo atlas
    uri = "mongodb://ac-wrb5fxy-shard-00-00.afy6vc5.mongodb.net:27017,ac-wrb5fxy-shard-00-01.afy6vc5.mongodb.net:27017,ac-wrb5fxy-shard-00-02.afy6v
    client_1 = MongoClient(uri, tls=True, tlsCertificateKeyFile='/content/X509-cert-1524735985430801619.pem')
```

```
[ ] #coleção dados da Fipe
    db = client_1['Projeto']
    colecao = db['FIPE-2022']
```

```
[ ] doc_count = colecao.count_documents({})
    print(doc_count)
```

```
[ ] # enviar o DF para colecao selecionada no mongo
    df2_dict = df2.to_dict("records")
    colecao.insert_many(df2_dict)
```

```
[ ] # escolha/crie o database e colecao
    db = client_1['Projeto']
    colecao = db['dadoss_finais']
```

```
[ ] dados_fipe_final_dict = dados_fipe_final.to_dict("records")
    colecao.insert_many(dados_fipe_final_dict)
```

```
[ ] colec = colecao.find({})
    dados_fipe = pd.DataFrame(list(colec))
```

```
[ ] colecao.count_documents({})
```

```
[ ] dados_fipe
```

Armazenamento dos dataframes resultantes na coleção do MongoDB

Envio da coleção Dados Brasil (ANFAVEA) para MongoDB

```
[ ] #coleção dados dataframe Brasil
    db2 = client_1['Projeto']
    colecao2 = db2['dados_brasil_consolidado']
```

```
[ ] doc_count = colecao2.count_documents({})
    print(doc_count)
```

```
0
```

```
[ ] # enviar o DF para colecao selecionada no mongo
    df3_dict = df3.to_dict('records')
    colecao2.insert_many(df3_dict)
```

```
<pymongo.results.InsertManyResult at 0x7fa30eb5b9a0>
```

```
colec_2 = colecao2.find({})
dados_brasil = pd.DataFrame(list(colec_2))
```

```
[ ] colecao2.count_documents({})
```

```
59
```

```
[ ] dados_brasil
```

	_id	data	vendas_saz	vendas_saz_preco_mes	vendas_saz_preco_ano	prod_auto_leve	prod_auto_leve_preco_atual_mes	pro
0	63c6b388557ee47cb29f0341	2021-08-01	141804.0	-4.51	-13.98	148857.0	0.70	
1	63c6b388557ee47cb29f0342	2018-02-01	175386.0	-0.52	8.93	203610.0	-3.30	

Análises com BigQuery

O BigQuery é o data warehouse corporativo da Google Cloud, o qual foi utilizado no projeto para fazer insights com consultas na linguagem SQL.

Principais marcas do Mercado Automotivo Brasileiro de 2018 até 2022

Quantidade de modelo

Marcas

Erro de configuração do conjunto de dados

Não foi possível conectar o Looker Studio ao seu conjunto de dados.

Ver detalhes

Tipo de câmbio

Erro de configuração do conjunto de dados

Não foi possível conectar o Looker Studio ao seu conjunto de dados.

Ver detalhes

Tipo de combustível

Erro de configuração do conjunto de dados

Não foi possível conectar o Looker Studio ao seu conjunto de dados.

Ver detalhes

FONTE: TABELA FIPE

Análise da potência dos motores de alguns veículos

Potência do motor de alguns modelos

Erro de configuração do conjunto de dados

Não foi possível conectar o Looker Studio ao seu conjunto de dados.

Ver detalhes

FONTE: TABELA FIPE

Quantidade de modelos dos veículos de 2018 até 2022

Ano

Quantidade de modelos por ano do veículo

Erro de configuração do conjunto de dados

Não foi possível conectar o Looker Studio ao seu conjunto de dados.

Ver detalhes

FONTE: TABELA FIPE

Média dos preços segundo a Tabela FIPE

Os modelos dos carros vão de 2013 até 2023

SELECIONE O MODELO

Preço médio por modelo dos veículos seguindo o ano 2022 como referência

Erro de configuração do conjunto de dados

Não foi possível conectar o Looker Studio ao seu conjunto de dados.

Ver detalhes

FONTE: TABELA FIPE

Análise dos carros para a verificação do maior preço médio

Variação de preço

Erro de configuração do

Erro de configuração do conjunto de dados

Não foi possível conectar o Looker Studio ao seu conjunto de dados.

Ver detalhes

FONTE: TABELA FIPE

Análise de produção e venda de automóveis e veículos comerciais leves

Erro de configuração do conjunto de dados

Erro de configuração do conjunto de dados

Erro de configuração do conjunto de dados

Não foi possível conectar o Looker Studio ao seu conjunto de dados.

Ver detalhes

Erro de configuração do conjunto de dados

Não foi possível conectar o Looker Studio ao seu conjunto de dados.

Ver detalhes

Cenário no período da pandemia de COVID-19

Produção e vendas de automóveis e veículos comerciais leves

Erro de configuração do conjunto de dados

Não foi possível conectar o Looker Studio ao seu conjunto de dados.

Ver detalhes

Comparativo

Comparação de vendas, produção e exportação de 2020 com 2019

Comparativo com 2019

Erro de configuração do conjunto de dados

Não foi possível conectar o Looker Studio ao seu conjunto de dados.

Ver detalhes

Comparativo com 2019

Erro de configuração do conjunto de dados

Não foi possível conectar o Looker Studio ao seu conjunto de dados.

Ver detalhes

Comparativo com 2019

Erro de configuração do conjunto de dados

Não foi possível conectar o Looker Studio ao seu conjunto de dados.

Ver detalhes

Mercado Automotivo

Produção / Vendas / Exportação

Erro de configuração do conjunto de dados

Não foi possível conectar o Looker Studio ao seu conjunto de dados.

Ver detalhes

Ano

Total vendas

Total produção

Total exportação

Conclusão dos dados analisados sobre o Mercado Automotivo

- **O Mercado Automotivo foi bastante afetado, mostrando a vulnerabilidade do setor frente as crises.**

- **O setor carece de muitos investimentos observado também pelo baixo número de exportação e o grande consumo de gasolina e diesel.**

NKVS Consultoria

SOUL CODE

Obrigado a todos!