

Allyson
O Analítico



Victor Hugo
O detalhista



Clério Fernandes
O curioso



Stéfanni Santos
A detetive



Thiago Breder
O comunicativo



Tomaz Silva
O estrategista

</Crimes(Seg . Pública)>

Projeto Final **SOUL CODE**

+

+

+

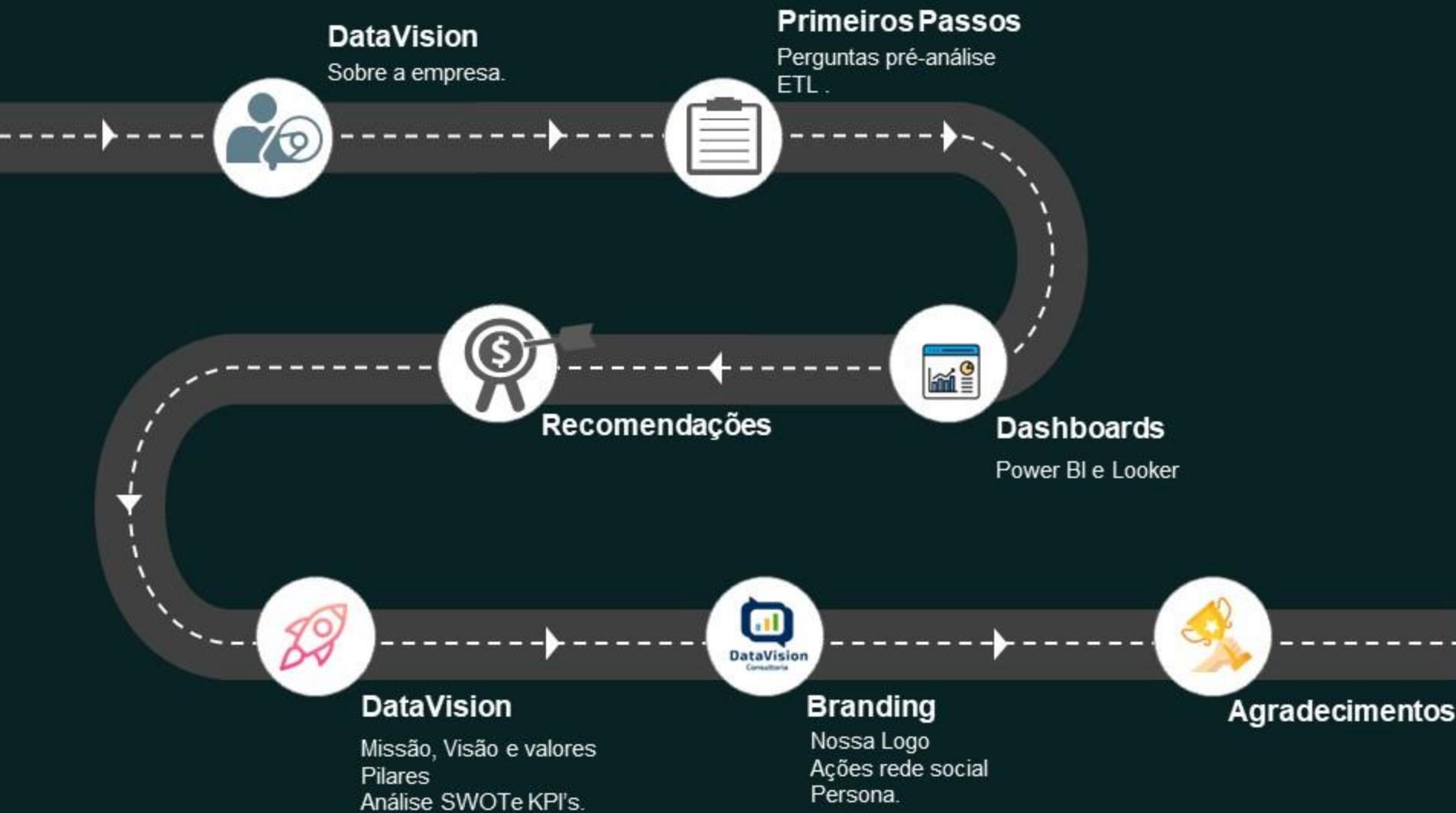
+



DataVision
Consultoria

+

ROADMAP >



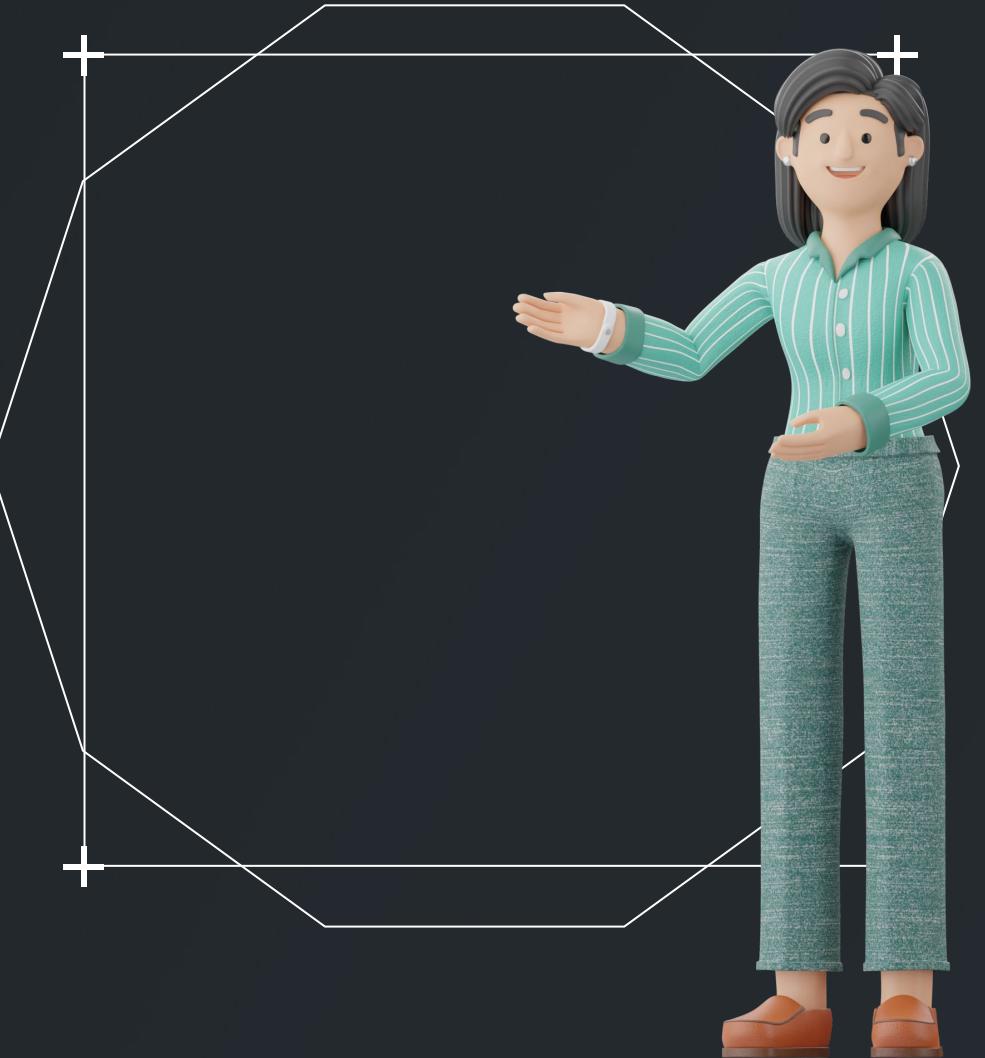
Quem <SOMOS>

Seja bem-vindo a DataVision Advisors!

Acreditamos que a análise de dados é fundamental para o sucesso dos negócios.

Somos especialistas em impulsionar o desempenho e o rendimento de nossos clientes.

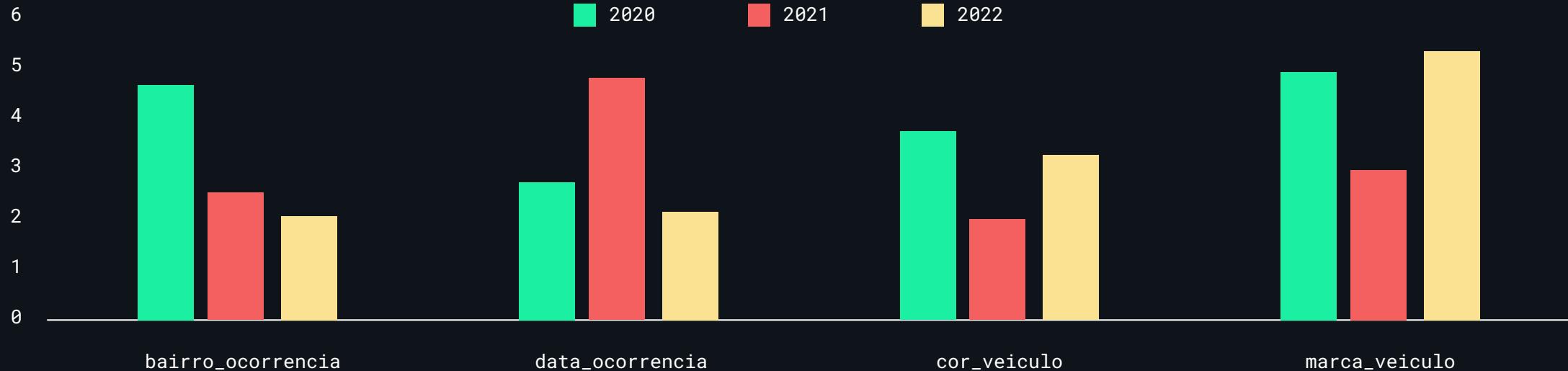
O nosso desejo é levar seus negócios para o próximo nível!



+ Campanha <SEGURANÇA PÚBLICA>



<Perguntas Pré-análise>



- Qual o número de ocorrências por ano?
- Em qual período do dia o número de furtos/roubos é maior?
- Em quais bairros acontecem mais furtos/roubos?

- Quais são os veículos mais furtados/roubados?
- O crescimento da frota de carros de SP foi afetado pelo Lockdown?
- O crescimento de casos de furtos/roubos é maior do que o volume de novos veículos?

Relatório Processo <ETL>

E

Nicho de pesquisa
Cidade de São Paulo

Fontes de dados;
IBGE - frota de carros
SECRETARIA DE SEGURANÇA DE SP. -
furtos e roubos registrados.
REUNIÃO COM A TEX - Índice de IPSA
SUSEP

- Avaliar o número de carros na cidade
- Comparar roubos e furtos por região
- IPSA para cruzar informações com os índices de criminalidade da SSP
- Verificar o perfil dos roubos e furtos efetuados.

T

Python - bibliotecas Pandas / Pyspark e MySQL(GCP)

Pandas - ferramenta mais usada, pois não trabalhamos com BigData.
Geramos algumas pré-visualizações nessa etapa, nos preparando para a etapa do Power BI e Looker.

- Muitas inconsistências no BD da SSP/SP, o que demandou um longo período de tratamento.
- Os outros BD tinham poucas inconsistências e não possuíam dados nulos, facilitando o processo.

L

Bucket GCP e MongoDB

Armazenamento dos dados tratados prontos para criação de Dashboards e análises no Bucket GCP e mongoDB.

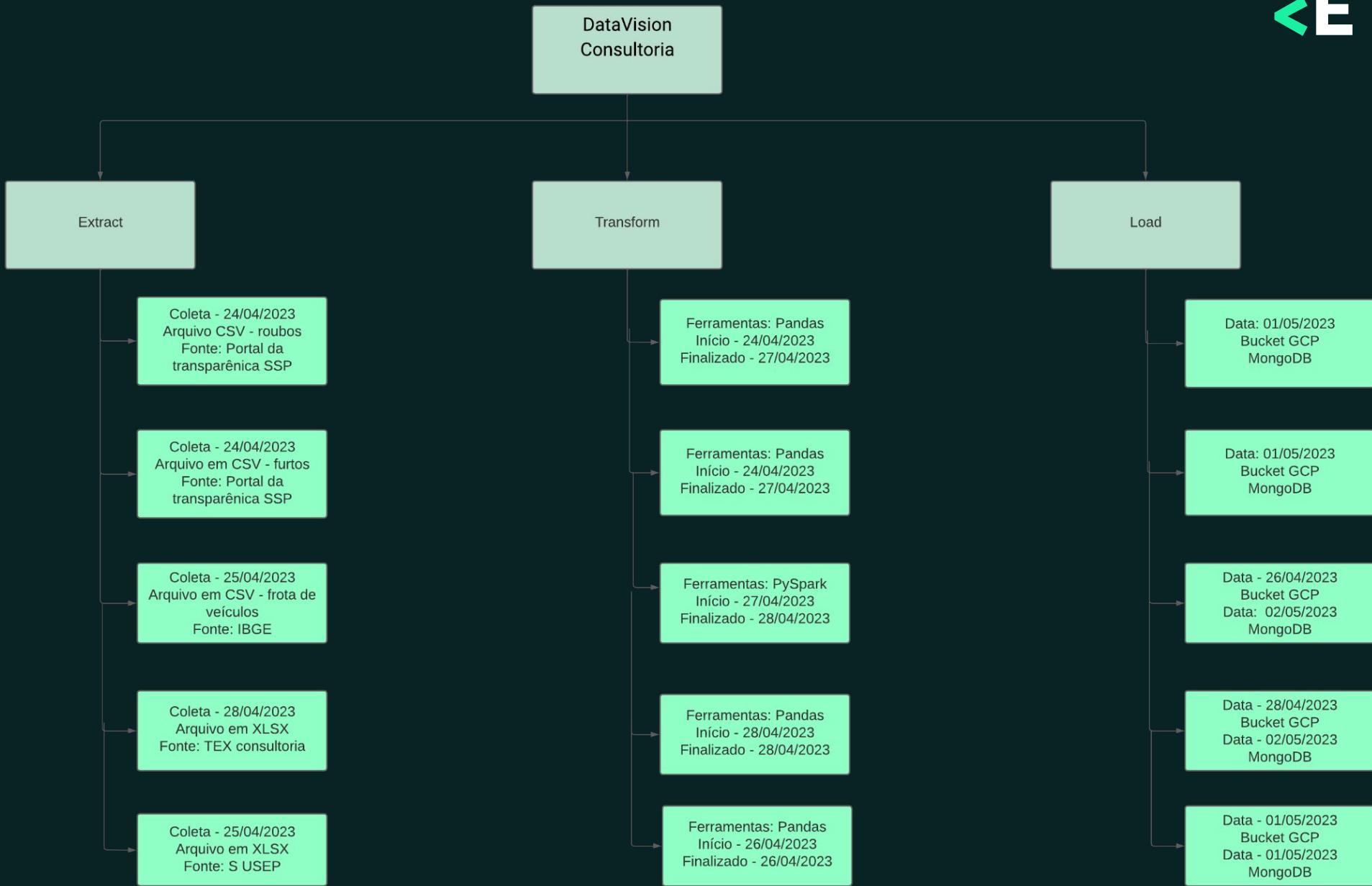
- Carregamento no bucket GCP sem intercorrências.
- Carregamento no mongoDB com tratamento de dados NA para NAN.



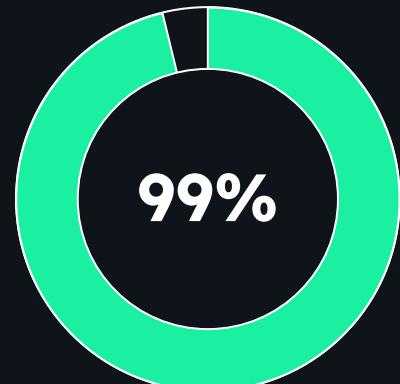
< W O R K F L O W >

<

<ETL>



<Ferramentas>



Conektor GCLOUD

```
[ ] #CONFIGURANDO DA CHAVE DE SEGURANCA (Enviada com o projeto)

serviceAccount = '/content/projeto-final-384719-f1efca397470.json'
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = serviceAccount

[ ] #Configurações Google Cloud Storage
client = storage.Client()
bucket = client.get_bucket('projeto_final_crimes')
```

```
[ ] #ROUBOS
path_roubos_2023 = 'gs://projeto_final_crimes/brutos_roubos/roubos_2023.csv'
path_roubos_2022 = 'gs://projeto_final_crimes/brutos_roubos/roubos_2022.csv'
path_roubos_2021 = 'gs://projeto_final_crimes/brutos_roubos/roubos_2021.csv'
path_roubos_2020 = 'gs://projeto_final_crimes/brutos_roubos/roubos_2020.csv'
path_roubos_2019 = 'gs://projeto_final_crimes/brutos_roubos/roubos_2019.csv'
path_roubos_2018 = 'gs://projeto_final_crimes/brutos_roubos/roubos_2018.csv'
```

Concatenação de Dataframes

Concatenando os dataframes

```
[ ] #ROUBOS  
df_roubos_2023 = pd.read_csv(path_roubos_2023,sep=',',encoding='utf-8')  
df_roubos_2022 = pd.read_csv(path_roubos_2022,sep=',',encoding='utf-8')  
df_roubos_2021 = pd.read_csv(path_roubos_2021,sep=',',encoding='utf-8')  
df_roubos_2020 = pd.read_csv(path_roubos_2020,sep=',',encoding='utf-8')  
df_roubos_2019 = pd.read_csv(path_roubos_2019,sep=',',encoding='utf-8')  
df_roubos_2018 = pd.read_csv(path_roubos_2018,sep=',',encoding='utf-8')
```

```
[ ] #FURTOS  
df_furtos_2023 = pd.read_csv(path_furtos_2023,sep=',',encoding='utf-8')  
df_furtos_2022 = pd.read_csv(path_furtos_2022,sep=',',encoding='utf-8')  
df_furtos_2021 = pd.read_csv(path_furtos_2021,sep=',',encoding='utf-8')  
df_furtos_2020 = pd.read_csv(path_furtos_2020,sep=',',encoding='utf-8')  
df_furtos_2019 = pd.read_csv(path_furtos_2019,sep=',',encoding='utf-8')  
df_furtos_2018 = pd.read_csv(path_furtos_2018,sep=',',encoding='utf-8')
```

```
[ ] df_furtos = pd.concat([df_furtos_2023,df_furtos_2022,df_furtos_2021,df_furtos_2020,df_furtos_2019,df_furtos_2018], axis=0).reset_index(drop=True)
```

```
[ ] df_roubos = pd.concat([df_roubos_2023,df_roubos_2022,df_roubos_2021,df_roubos_2020,df_roubos_2019,df_roubos_2018], axis=0).reset_index(drop=True)
```

+

Dropando linhas e colunas

Analisando os dados do ano de 2023, percebemos muitas inconsistências, como colunas trocadas e duas colunas a mais vazias, além do fato de ter somente dois meses disponíveis para análise, portanto, decidimos dropar o ano 2023.

```
[ ] df_roubos = pd.concat([df_roubos_2022,df_roubos_2021,df_roubos_2020,df_roubos_2019,df_roubos_2018], axis=0).reset_index(drop=True)
df_furtos = pd.concat([df_furtos_2022,df_furtos_2021,df_furtos_2020,df_furtos_2019,df_furtos_2018], axis=0).reset_index(drop=True)
```

As colunas NUM_BO, NUMERO_BOLETIM, BO_INICIADO, BO_EMITIDO, DATACOMUNICACAO, DATAELABORACAO, BO_AUTORIA, NUMERO_BOLETIM_PRINCIPAL, EXAME, SOLUCAO, PLACA_VEICULO, DELEGACIA_NOME, DELEGACIA_CIRCUNSCRICAO, ESPECIE, RUBRICA, DESDOBRAMENTO, TIPOPESSOA, VITIMAFATAL, NATURALIDADE, NACIONALIDADE, SEXO, IDADE, DATANASCIMENTO, IDADE, ESTADOCIVIL, PROFISSAO, GRAUINSTRUCAO, CORCUTIS, NATUREZAVINCULADA, TIPOVINCULO, RELACIONAMENTO, PARENTESCO, PLACAVEICULO, QUANT_CELULAR, MARCA_CELULAR ou por estarem vazios ou porque não serão úteis para análise proposta.

```
[ ] df_roubos.drop(['NUM_BO', 'NUMERO_BOLETIM', 'BO_INICIADO', 'BO_EMITIDO', 'DATACOMUNICACAO',
                   'DATAELABORACAO', 'BO_AUTORIA', 'NUMERO_BOLETIM_PRINCIPAL', 'EXAME',
                   'SOLUCAO', 'PLACA_VEICULO', 'DELEGACIA_NOME', 'DELEGACIA_CIRCUNSCRICAO',
                   'ESPECIE', 'RUBRICA', 'DESCOBERTO', 'TIPOPESSOA',
                   'VITIMAFATAL', 'NATURALIDADE', 'NACIONALIDADE', 'SEXO',
                   'IDADE', 'DATANASCIMENTO', 'ESTADOCIVIL', 'PROFISSAO',
                   'GRAUINSTRUCAO', 'CORCUTIS', 'NATUREZAVINCULADA', 'TIPOVINCULO',
                   'RELACIONAMENTO', 'PARENTESCO', 'PLACAVEICULO', 'QUANT_CELULAR',
                   'MARCA_CELULAR'], axis=1, inplace=True)
```

Conversão de Formatos

```
[ ] for data in df_roubos['DATAOCORRENCIA']:
    try:
        pd.to_datetime(data, format='%d/%m/%Y')
    except Exception as e:
        print(f"Erro ao converter {data}: {e}")

Erro ao converter 17/12/0202: Out of bounds nanosecond timestamp: 202-12-17 00:00:00
Erro ao converter 23/01/1019: Out of bounds nanosecond timestamp: 1019-01-23 00:00:00
Erro ao converter 02/03/1019: Out of bounds nanosecond timestamp: 1019-03-02 00:00:00

[ ] df_roubos.DATAOCORRENCIA.replace(to_replace ="17/12/0202",value = '17/12/2020', regex=True,inplace=True)
df_roubos.DATAOCORRENCIA.replace(to_replace ="23/01/1019",value = '23/01/2019', regex=True,inplace=True)
df_roubos.DATAOCORRENCIA.replace(to_replace ="02/03/1019",value = '02/03/2019', regex=True,inplace=True)

[ ] df_roubos['DATAOCORRENCIA'] = pd.to_datetime(df_roubos['DATAOCORRENCIA'], format='%d/%m/%Y')

[ ] for data in df_roubos['HORAOCORRENCIA']:
    try:
        pd.to_datetime(data, format='%H:%M')
    except Exception as e:
        print(f"Erro ao converter {data}: {e}")

[ ] df_roubos['HORAOCORRENCIA']= pd.to_datetime(df_roubos['HORAOCORRENCIA'],format='%H:%M')

[ ] df_roubos['NUMERO'] = df_roubos['NUMERO'].astype(str)

[ ] df_roubos['ANO_FABRICACAO']=df_roubos.ANO_FABRICACAO.round().astype('Int64')

[ ] df_roubos['ANO_MODELO']=df_roubos.ANO_MODELO.round().astype('Int64')
```

Criando uma nova coluna

```
▶ #Filtro para letras dentro do valor
filtrointerno = df_roubos['DESCR_MARCA_VEICULO'].fillna('').str.contains('HONDA/CG150')
df_roubos.loc[filtrointerno].groupby(['DESCR_MARCA_VEICULO']).size().sort_values(ascending=False)

□ DESCR_MARCA_VEICULO
HONDA/CG150 FAN ESDI    768
HONDA/CG150 START      104
HONDA/CG150 TITAN MIX KS 100
HONDA/CG150 CARGO ESD     87
HONDA/CG150FANCARGO ESDI   64
HONDA/CG150 TITAN MIX EX    63
HONDA/CG150 TITAN MIX ES    47
HONDA/CG150 TITAN MIXESD   32
HONDA/CG150 SPECIAL EDIT     1
dtype: int64
```

```
[ ] df_roubos['modelo'] = df_roubos['DESCR_MARCA_VEICULO'].str.split(' ', n=2, expand=True)[0] + ' ' + df_roubos['DESCR_MARCA_VEICULO'].str.split(' ', n=2, expand=True)[1]
df_roubos['modelo'].head(5)

0      I/PEUGEOT 307
1      FORD/KA SE
2      I/HYUNDAI TUCSON
3      FIAT/PALIO ELX
4      NISSAN/KICKS ACTIVE
Name: modelo, dtype: object

[ ] df_roubos['modelo'] = np.where(df_roubos['modelo'].str.startswith('I/'), df_roubos['modelo'], df_roubos['modelo'].str.split(' ', n=2, expand=True)[0])
df_roubos['modelo'] = df_roubos['modelo'].fillna(df_roubos['DESCR_MARCA_VEICULO'])
df_roubos['modelo'].head(5)

0      I/PEUGEOT 307
1      FORD/KA
2      I/HYUNDAI TUCSON
3      FIAT/PALIO
4      NISSAN/KICKS
Name: modelo, dtype: object
```

Criando uma nova coluna

```
[ ] df_roubos.groupby(['modelo']).size().sort_values(ascending=False).head(20)

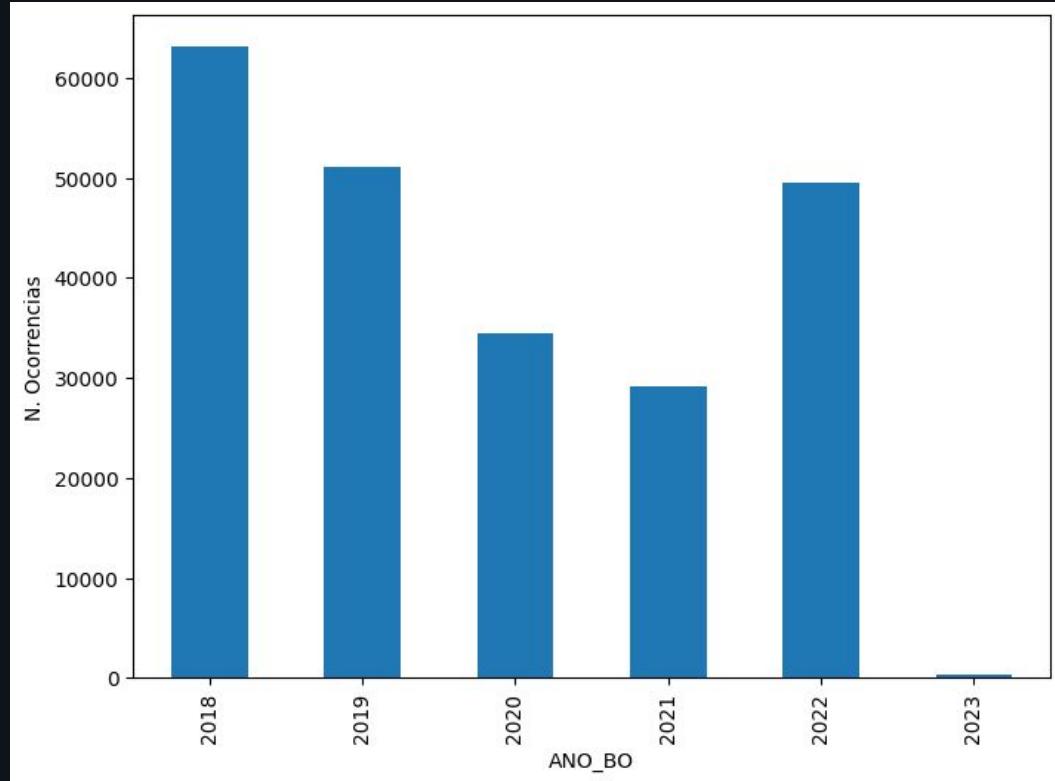
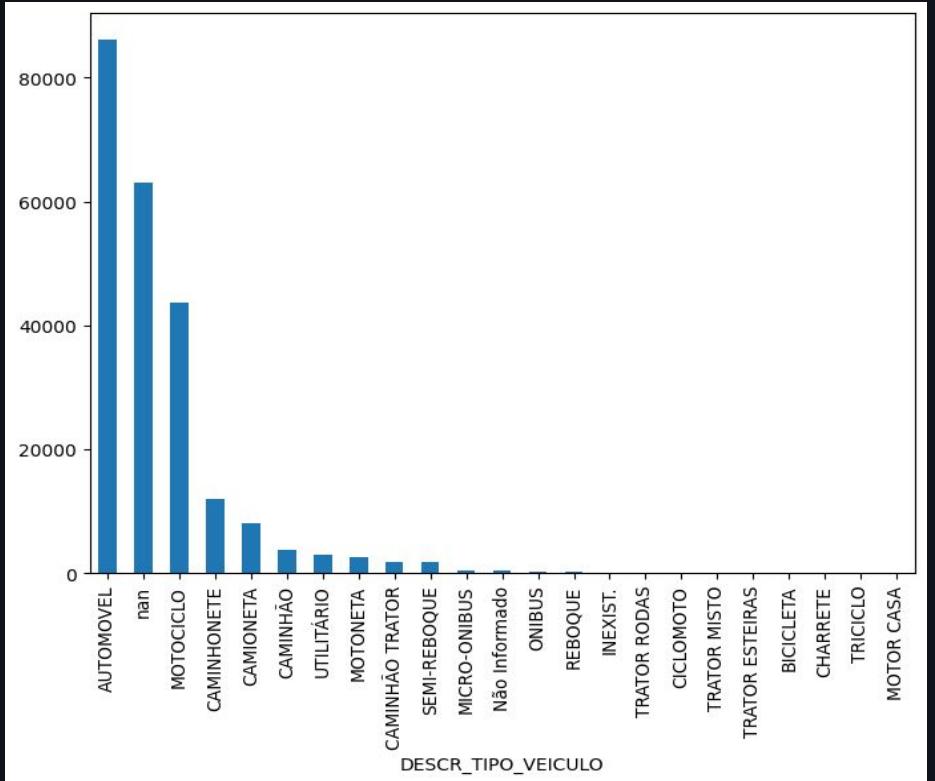
modelo
HONDA/CG           10823
CHEVROLET/ONIX     4122
FORD/KA            3603
HYUNDAI/HB20       3380
HONDA/FIT          3341
HONDA/CB           2898
RENAULT/SANDERO    2786
HONDA/XRE          2752
FIAT/PALIO         2695
FIAT/FIORINO       2488
FIAT/UNO           2377
FORD/FIESTA        2276
CHEV/PRISMA         2108
VW/FOX              2106
TOYOTA/COROLLA     2056
HONDA/PCX           1973
RENAULT/LOGAN       1914
HONDA/CIVIC         1798
JEEP/RENEGADE       1701
YAMAHA/XTZ250       1659
dtype: int64
```

Por praticidade, decidimos trabalhar com os 20 modelos mais roubados/furtados, porque para tratar as inconsistências vindas do banco de dados, exigiria muito mais tempo dedicado ao tratamento de dados

```
[ ] #Corrigindo inconsistência do ONIX
df_roubos.modelo.replace(to_replace ="CHEV/ONIX",value = 'CHEVROLET/ONIX', regex=True,inplace=True)
```

```
[ ] # CHEVROLET PRISMA
df_roubos.modelo.replace(to_replace ="CHEV/PRISMA",value = 'CHEVROLET/PRISMA', regex=True,inplace=True)
df_roubos.modelo.replace(to_replace ="GM/PRISMA",value = 'CHEVROLET/PRISMA', regex=True,inplace=True)
```

Visualizações



Renomeando colunas

```
[ ] df_furtos.columns  
  
Index(['ANO_BO', 'DATAOCORRENCIA', 'HORAOCORRENCIA', 'PERIDOOCORRENCIA',  
       'FLAGRANTE', 'LOGRADOURO', 'NUMERO', 'BAIRRO', 'CIDADE', 'UF',  
       'LATITUDE', 'LONGITUDE', 'DESCRICAOLOCAL', 'STATUS', 'UF_VEICULO',  
       'CIDADE_VEICULO', 'DESCR_COR_VEICULO', 'DESCR_MARCA_VEICULO',  
       'ANO_FABRICACAO', 'DESCR_TIPO_VEICULO', 'modelo'],  
       dtype='object')  
  
[ ] df_roubos.rename(columns={'ANO_BO': 'ANO', 'DATAOCORRENCIA': 'DATA DA OCORRÊNCIA',  
                           'PERIDOOCORRENCIA': 'PERÍODO DO DIA', 'NUMERO': 'NÚMERO',  
                           'DESCRICAOLOCAL': 'DESCRIPAÇÃO DO LOCAL', 'UF_VEICULO': 'UF DO VEÍCULO',  
                           'CIDADE_VEICULO': 'CIDADE DO VEÍCULO', 'DESCR_COR_VEICULO': 'COR DO VEÍCULO',  
                           'DESCR_MARCA_VEICULO': 'VERSÃO DO VEÍCULO', 'ANO_FABRICACAO': 'FABRICAÇÃO',  
                           'DESCR_TIPO_VEICULO': 'TIPO DE VEÍCULO', 'modelo': 'MODELO DO VEÍCULO'}, inplace=True)  
  
df_furtos.rename(columns={'ANO_BO': 'ANO', 'DATAOCORRENCIA': 'DATA DA OCORRÊNCIA',  
                           'PERIDOOCORRENCIA': 'PERÍODO DO DIA', 'NUMERO': 'NÚMERO',  
                           'DESCRICAOLOCAL': 'DESCRIPAÇÃO DO LOCAL', 'UF_VEICULO': 'UF DO VEÍCULO',  
                           'CIDADE_VEICULO': 'CIDADE DO VEÍCULO', 'DESCR_COR_VEICULO': 'COR DO VEÍCULO',  
                           'DESCR_MARCA_VEICULO': 'VERSÃO DO VEÍCULO', 'ANO_FABRICACAO': 'FABRICAÇÃO',  
                           'DESCR_TIPO_VEICULO': 'TIPO DE VEÍCULO', 'modelo': 'MODELO DO VEÍCULO'}, inplace=True)
```



MongoDB

```
[ ] import pymongo
from pymongo import MongoClient

uri = "mongodb+srv://tomaz03.0fildnq.mongodb.net/?authSource=%24external&authMechanism=MONGODB-X509&retryWrites=true&w=majority"
client_mongo = MongoClient(uri,
                           tls=True,
                           tlsCertificateKeyFile='/content/X509-cert-3463438694104533343.pem')

db = client_mongo['Tratados']
colecao = db['roubos']
```

```
[ ] df_roubos = df_roubos.astype(object).where(df_roubos['HORAOCORRENCIA'].notnull(), None)
df_roubos.fillna('', inplace=True)
colecao.insert_many(df_roubos.to_dict("records"))

<pymongo.results.InsertManyResult at 0x7fb8bbdb8ca0>
```

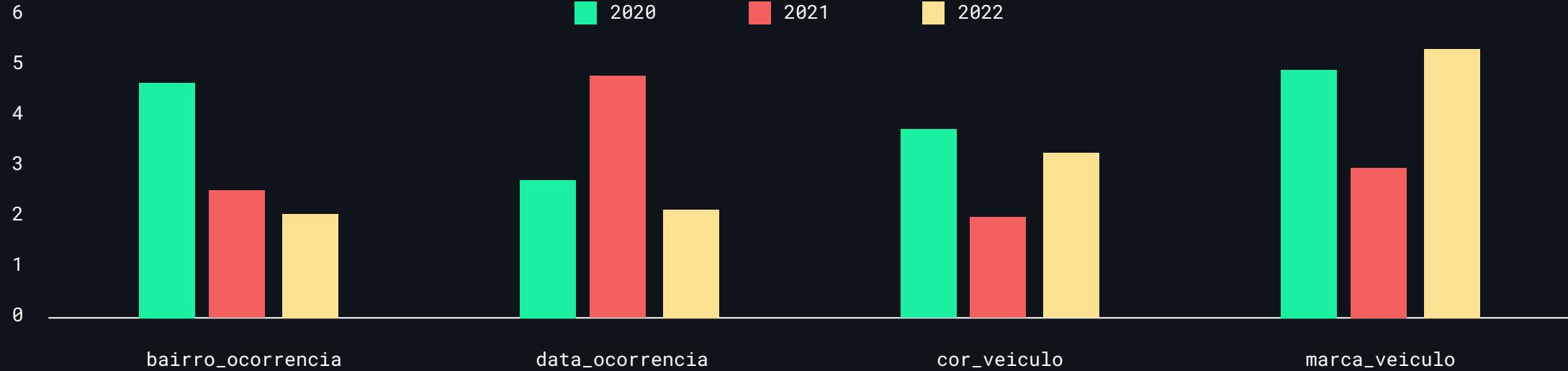
Consultas <BIGQUERY>

```
1 SELECT MODELO_DO_VE__CULO, COUNT(*) AS CONTAGEM
2 FROM `projeto-final-384719.projeto_final_dados_crimes.roubos_tratados` WHERE MODELO_DO_VE__CULO IS NOT NULL
3 GROUP BY MODELO_DO_VE__CULO
4 ORDER BY CONTAGEM DESC LIMIT 20;
5
6 SELECT BAIRRO, COUNT(*) AS CONTAGEM
7 FROM `projeto-final-384719.projeto_final_dados_crimes.roubos_tratados` WHERE BAIRRO IS NOT NULL
8 GROUP BY BAIRRO
9 ORDER BY CONTAGEM DESC LIMIT 10;
10
11 SELECT PERODO_DO_DIA, COUNT(*) AS CONTAGEM
12 FROM `projeto-final-384719.projeto_final_dados_crimes.roubos_tratados` WHERE PERODO_DO_DIA IS NOT NULL
13 GROUP BY PERODO_DO_DIA
14 ORDER BY CONTAGEM DESC;
15
16 SELECT DESCRI___O_DO_LOCAL, COUNT(*) AS CONTAGEM
17 FROM `projeto-final-384719.projeto_final_dados_crimes.roubos_tratados` WHERE DESCRI___O_DO_LOCAL IS NOT NULL
18 GROUP BY DESCRI___O_DO_LOCAL
19 ORDER BY CONTAGEM DESC LIMIT 10;
```

Consultas <BIGQUERY>

```
1 SELECT MODELO_DO_VE_CULO, COUNT(*) AS CONTAGEM
2 FROM `projeto-final-384719.projeto_final_dados_crimes.furtos_tratados` WHERE MODELO_DO_VE_CULO IS NOT NULL
3 GROUP BY MODELO_DO_VE_CULO
4 ORDER BY CONTAGEM DESC LIMIT 20;
5
6 SELECT BAIRRO, COUNT(*) AS CONTAGEM
7 FROM `projeto-final-384719.projeto_final_dados_crimes.furtos_tratados` WHERE BAIRRO IS NOT NULL
8 GROUP BY BAIRRO
9 ORDER BY CONTAGEM DESC LIMIT 10;
10
11 SELECT PER_ODO_DO_DIA, COUNT(*) AS CONTAGEM
12 FROM `projeto-final-384719.projeto_final_dados_crimes.furtos_tratados` WHERE PER_ODO_DO_DIA IS NOT NULL
13 GROUP BY PER_ODO_DO_DIA
14 ORDER BY CONTAGEM DESC;
15
16 SELECT DESCRI_O_DO_LOCAL, COUNT(*) AS CONTAGEM
17 FROM `projeto-final-384719.projeto_final_dados_crimes.furtos_tratados` WHERE DESCRI_O_DO_LOCAL IS NOT NULL
18 GROUP BY DESCRI_O_DO_LOCAL
19 ORDER BY CONTAGEM DESC LIMIT 10;
20
```

<DASHBOARDS>



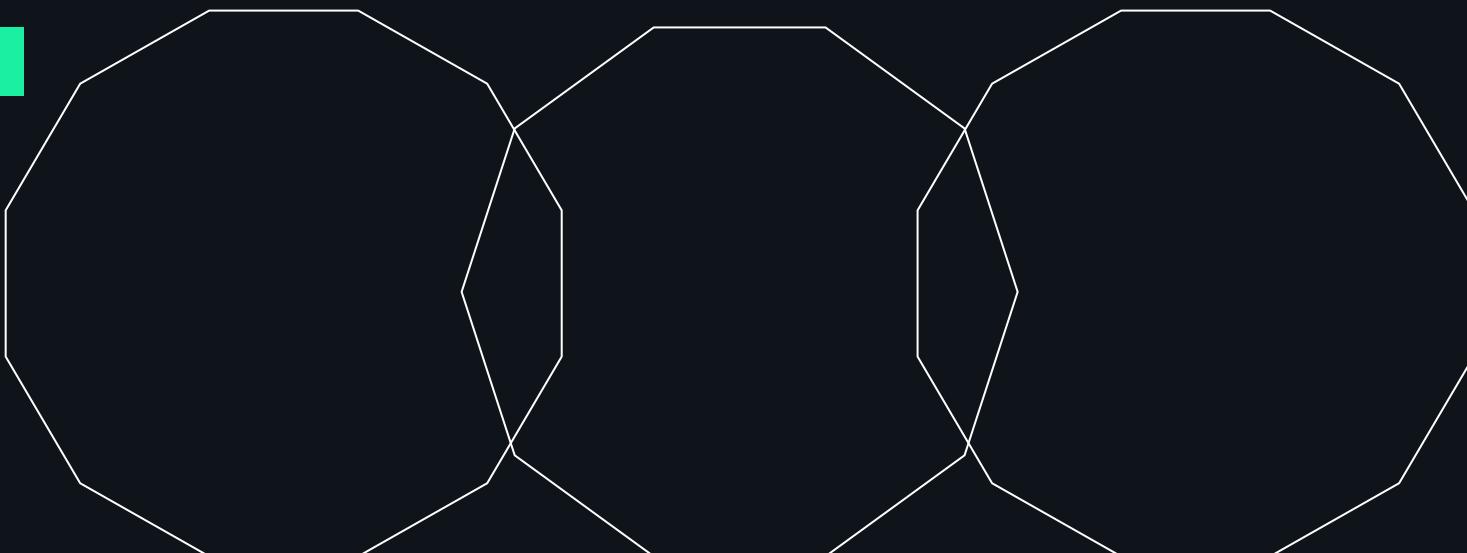
Power BI



 **Looker**

RECOMENDAÇÕES:

- Elaborar perguntas mais estratégicas durante a análise de perfil, para verificar quais clientes são mais propensos a deixar seus veículos expostos em vias públicas;
- Fazer parcerias com estacionamentos , oferecer descontos, programas de pontos, brindes etc.;
- Utilizar o app da seguradora para enviar mensagens notificando o cliente quando ele estiver em um bairro com alto índice de furtos/roubos de veículos;
- Enviar alertas comunicando o cliente sobre os horários de maior índice de furtos/roubos de veículos em determinada região.;
- Contratar o serviço especializado da DataVision para ajudar a sua empresa a reduzir o índice de sinistralidade.



```
< Product details={data}>  
/>  
< /Layout >
```





DataVision

Consultoria

VALORES

INOVAÇÃO
QUALIDADE
CONFIANÇA
INTEGRIDADE
COLABORAÇÃO



MISSÃO

Fornecer soluções confiáveis, que ajudem nossos clientes a tomar decisões inteligentes e estratégicas.

VISÃO

Ser a empresa líder em análise de dados e referência no uso de tecnologias inovadoras.

+

Pilares <DATAVISION>



<Análise SWOT>



FORÇAS

- Conhecimento especializado em análise de dados para seguradora de veículos.
- Profissionais experientes e comprometidos no processo de coleta, organização e análise de grandes conjuntos de dados.



FRAQUEZAS

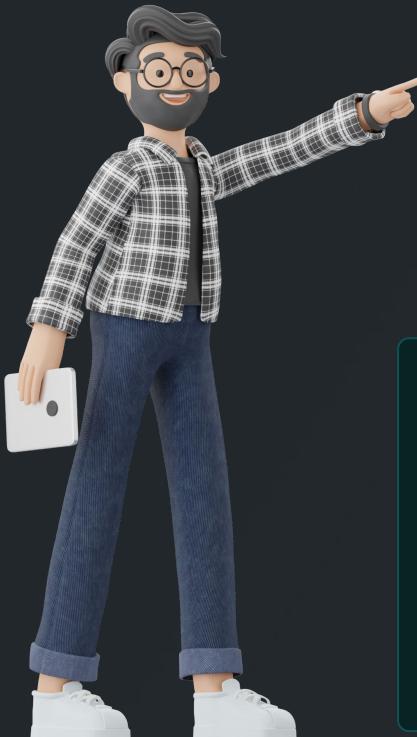
- Dificuldade em acessar fontes de dados completas em órgãos públicos.
- Complexidade em atrair novos clientes, especialmente em um mercado altamente competitivo.

**ESSA ANÁLISE
É REAL ?**



<Análise SWOT>

VEM NOVIDADE POR AÍ!



ROUBOS E FURTOS DE CELULARES SÃO PAULO/SP



2022 => 202.027
2021 => 179.335
2020 => 173.610

Fonte:
<https://g1.globo.com/monitor-da-violencia/noticia/2023/03/26/cidade-de-sp-tem-um-registro-de-celular-roubado-ou-furtado-a-cada-3-minutos.ghtml>



OPORTUNIDADES

- Possibilidade de expandir nossa atuação para outros setores do mercado de seguros.
- Novas tendências e tecnologias em análise de dados para oferecer soluções inovadoras aos nossos clientes.

AMEAÇAS

- Concorrência.
- Mudanças regulatórias no setor.
- Possíveis problemas de segurança.

<Principais métricas e KPI's>



+

+

+

+

////

- RECEITA
- TAXA DE CONVERSÃO
- RETENÇÃO DE CLIENTES
- SATISFAÇÃO DO CLIENTE
- NÚMERO DE PROJETOS CONCLUÍDOS

<BRANDING>



Campanhas <LINKEDIN>

+

VOCÊ SABE QUEM
VOCÊ É, MAS NÃO
SABE QUEM PODE
SE TORNAR

Seja destaque do seu setor com os insights certos que
seus dados podem fornecer !



DESTAQUE SEU NEGÓCIO
COM DADOS !



ALCANCE MELHORES
RESULTADOS COM
DADOS



Nossa <LOGO>

Cores da marca

O gráfico crescente representa a nossa evolução constante.

Uma empresa orgulhosamente brasileira, carregamos as cores da nossa bandeira.



Balão para diálogo.

Somos uma empresa que valoriza a comunicação !

DataVision

Nossa VISÃO de dados vai além de uma simples visualização !

DataVision

Consultoria

Mentores

Somos um time com o mindset focado em gerar valor através dos dados.

Nossa <MARCA>



DataVision
Consultoria

Conheça o **<JOHNY>**

O nosso buyer persona !



DataVision
Consultoria

⁺<Buyer Persona>

COMPORTAMENTAIS:

- HÁBITOS DE COMPRAS
- PREFERÊNCIAS DE CONSUMO
- MOTIVAÇÕES
- EXPECTATIVAS

+



+

<Buyer Persona>

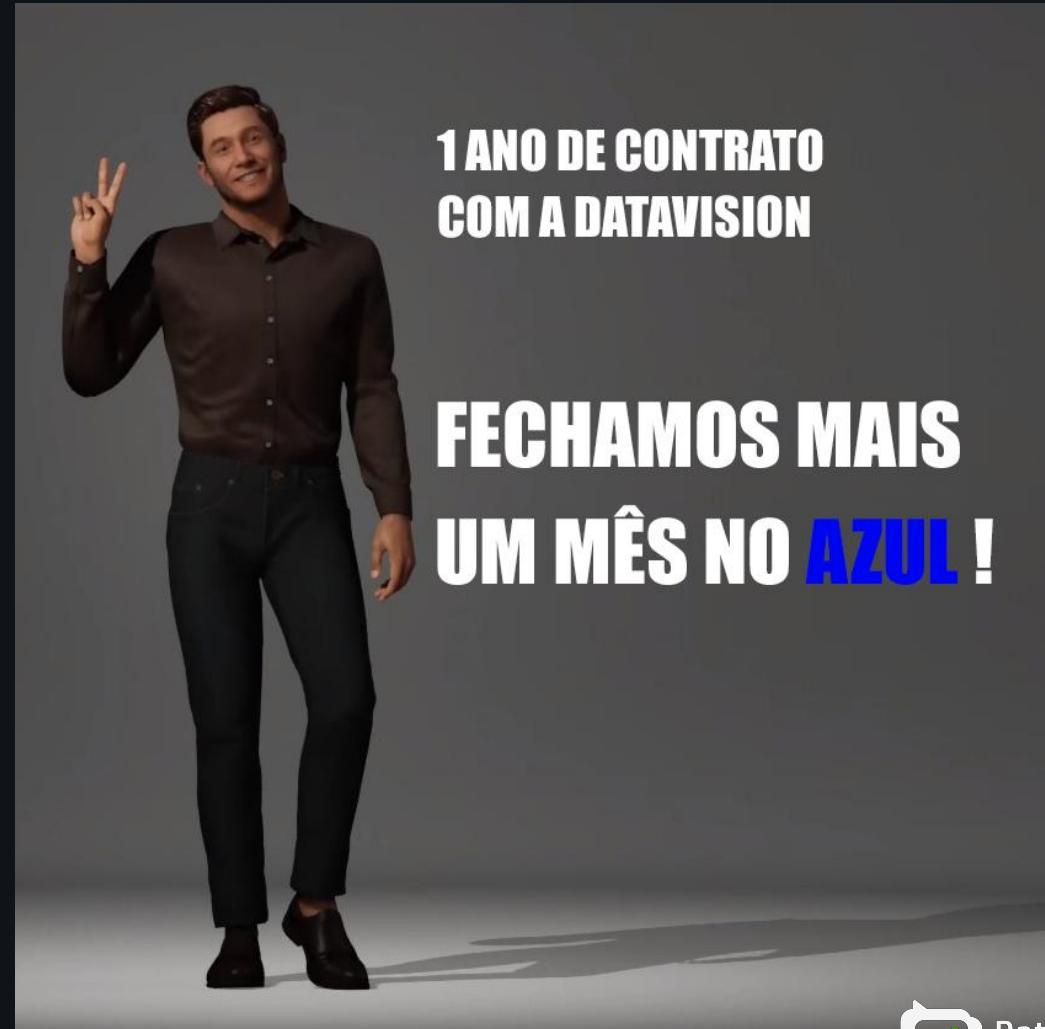


USO DO PRODUTO E SERVIÇO:

- PORQUE UTILIZA O SERVIÇO
- COMO UTILIZA O SERVIÇO
- FREQUÊNCIA DE USO
- BENEFÍCIOS VALORIZADOS
- CARACTERÍSTICAS DESEJADAS



<Buyer Persona>



soul code

+

meio&mensagem

<Obrigado(a)>

+

 **Santander**

Martech Academy

“Nenhum dever é mais importante do que a gratidão.”
Cícero

+

