

## MVP – Minimum Viable Product

### Pipeline Engenharia de Dados

## ÍNDICE

### Sumário

Controle de Versão .....	3
1. OBJETIVO .....	4
2. PLATAFORMA .....	4
3. DETALHAMENTO .....	4
3.1 BUSCA PELOS DADOS .....	4
3.2 COLETA.....	5
3.3 MODELAGEM .....	6
CATÁLOGO DOS DADOS:.....	7
LINHAGEM DOS DADOS: .....	7
4. CARGA .....	8
5. ANALISE .....	9
6. ARQUITETURA .....	12
7. FLUXO PIPELINE .....	13
8. AUTOAVALIAÇÃO.....	19

## Controle de Versão

Nessa seção, serão descritas todas as atualizações realizadas no documento, bem como a data e o responsável por elas.

DATA	VERSÃO	RESPONSÁVEL	ATUALIZAÇÃO
12/06/2024	1.0	LEONARDO SILVA	LAYOUT DOCUMENTO
13/06/2024	1.1	LEONARDO SILVA	ESTUDO SOBRE O TEMA.
16/06/2024	1.2	LEONARDO SILVA	ÍNDICE, DESCRIÇÃO INICIAL.
22/06/2024	1.3	LEONARDO SILVA	FERRAMENTAS WEB GCP
01/07/2024	1.4	LEONARDO SILVA	CONFIG METABASE
04/07/2024	1.5	LEONARDO SILVA	CATALOGO DE DADOS
06/07/2024	1.5	LEONARDO SILVA	FERRAMENTAS WEB BIGQUERY / DATAFLOW
07/07/2024	1,6	LEONARDO SILVA	IMAGENS E ESTRUTURA DE ENTREGA
08/07/2024	1.7	LEONARDO SILVA	EXECUÇÃO FLUXO PIPELINE E IMAGENS
10/07/2024	1.8	LEONARDO SILVA	REVISÃO E AUTOAVALIAÇÃO
12/07/2024	1.9	LEONARDO SILVA	ENTREGA

## 1. OBJETIVO

O projeto tem o objetivo desenvolver uma pipeline de dados utilizando tecnologias na nuvem. Essa pipeline deve necessariamente envolver a busca, coleta, modelagem, carga e análise dos dados, bem como responder 5 perguntas relacionadas a emissão de CNH's (Carteira Nacional de Habilitação), emitidas em 4 estados brasileiros, com base em uma volumetria/coleta de dados por período.

Perguntas:

- 1- Total Emissão por UF.
- 2- Porcentagem % Emissão por UF.
- 3- Porcentagem % de emissão por TIPO CNH. (Permissionário x Condutor).
- 4- Media tipo categoria por idade.
- 5- Media idade primeira habilitação.

## 2. PLATAFORMA

Direcionei os esforços de apoio na Plataforma **Dataprocc**. O Dataprocc é um serviço de nuvem rápido, fácil de usar e totalmente gerenciado para executar clusters do Apache Spark e Apache Hadoop com mais simplicidade. O Databricks Community Edition sugerido, possui uma particularidade na versão não muito amigável que tende a ficar recriando a todo momento a pipeline.

## 3. DETALHAMENTO

### 3.1 BUSCA PELOS DADOS

Iniciei um garimpo na busca por dataset's gratuitos disponíveis na web, identifiquei alguns bons e interessantes, cheguei a iniciar uma análise dos benefícios por localidade (<https://portaldatransparencia.gov.br/beneficios>), porem me senti mais confortável em usar uma pequena massa de dados relacionado a emissão de CNH de 4 estados brasileiros.

**\*\* Dados privados tratados devido LGPD.**

## 3.2 COLETA

A coleta inicial dos dados foi feita em uma base de dados SqlServer, composta pela consulta abaixo, trazendo dados de forma original e posteriormente exportados para um arquivo .csv.

SELECT \* FROM emissao\_cnh c where c.STATUS = 22;

Clique >> [Dados Inicial](#)

Resultados		Mensagens								
	SIT_INT_ID_SITE	CAR_INT_CD_STATUS	CAR_STR_NR_CPF	CAR_STR_DS_NOMEINHA1	CAR_DAT_DT_NASCIMENTO	CAR_STR_ID_CATEGORIA	CAR_STR_ID_TIPO	CAR_STR_DS_UF	CAR_DAT_DT_EMISSAO	CAR_DAT_DT_
1	8	22	5185XXX380	LEMILITO SILVA NASCIME	1991-07-11 00:00:00.000000 UTC	AB	C	MA	2024-05-07 00:00:00.000000 UTC	2012-02-13 00:00:00.000000 UTC
2	8	22	5529XXX383	CARLIANE CUNHA SILVA	1990-12-01 00:00:00.000000 UTC	AB	P	MA	2024-05-07 00:00:00.000000 UTC	2024-05-07 00:00:00.000000 UTC
3	8	22	1813XXX374	SAYMON STHEVANO FIGUEI	2000-03-20 00:00:00.000000 UTC	B	C	MA	2024-05-07 00:00:00.000000 UTC	2018-11-14 00:00:00.000000 UTC
4	8	22	1098XXX8356	SERGIO SOUSA SILVA	2004-04-29 00:00:00.000000 UTC	AB	P	MA	2024-05-07 00:00:00.000000 UTC	2024-05-07 00:00:00.000000 UTC
5	8	22	7858XXX5372	JULIO CESAR SOARES NAS	1980-07-23 00:00:00.000000 UTC	AE	C	MA	2024-05-07 00:00:00.000000 UTC	2001-05-21 00:00:00.000000 UTC
6	8	22	6978XXX4353	WALDEMIR SILVA COSTA	1973-08-14 00:00:00.000000 UTC	AD	C	MA	2024-05-07 00:00:00.000000 UTC	2002-07-09 00:00:00.000000 UTC
7	8	22	6978XXX4353	WALDEMIR SILVA COSTA	1973-08-14 00:00:00.000000 UTC	AD	C	MA	2024-05-07 00:00:00.000000 UTC	2002-07-09 00:00:00.000000 UTC
8	8	22	9216XXX6300	CELIO DE BARROS VAZ	1983-06-27 00:00:00.000000 UTC	AD	C	MA	2024-05-07 00:00:00.000000 UTC	2009-12-29 00:00:00.000000 UTC
9	8	22	6027XXX8301	WANDERSON RODRIGUES DA	1990-03-29 00:00:00.000000 UTC	AB	C	MA	2024-05-07 00:00:00.000000 UTC	2019-11-21 00:00:00.000000 UTC
10	8	22	2978XXX1387	MARCELINO DE ARAUJO SI	1966-06-02 00:00:00.000000 UTC	AD	C	MA	2024-05-07 00:00:00.000000 UTC	2006-04-25 00:00:00.000000 UTC
11	8	22	2978XXX1387	MARCELINO DE ARAUJO SI	1966-06-02 00:00:00.000000 UTC	AD	C	MA	2024-05-07 00:00:00.000000 UTC	2006-04-25 00:00:00.000000 UTC
12	8	22	627XXX92382	MARCOS KAUÁ SILVA CORD	2003-05-10 00:00:00.000000 UTC	AB	P	MA	2024-05-07 00:00:00.000000 UTC	2024-05-07 00:00:00.000000 UTC
13	8	22	6215XXX0318	EDUARDO SILVA NASCIME	2004-08-08 00:00:00.000000 UTC	AB	P	MA	2024-05-07 00:00:00.000000 UTC	2024-05-07 00:00:00.000000 UTC
14	8	22	6215XXX0318	EDUARDO SILVA NASCIME	2004-08-08 00:00:00.000000 UTC	AB	P	MA	2024-05-07 00:00:00.000000 UTC	2024-05-07 00:00:00.000000 UTC
15	8	22	3811XXX4204	ALESSANDRO SANTANA CHA	1975-04-21 00:00:00.000000 UTC	AE	C	MA	2024-05-07 00:00:00.000000 UTC	1995-03-23 00:00:00.000000 UTC
16	8	22	3811XXX4204	ALESSANDRO SANTANA CHA	1975-04-21 00:00:00.000000 UTC	AE	C	MA	2024-05-07 00:00:00.000000 UTC	1995-03-23 00:00:00.000000 UTC
17	8	22	3237XXX4899	ANTONIO FRANCISCO DA S	1984-01-16 00:00:00.000000 UTC	AE	C	MA	2024-05-07 00:00:00.000000 UTC	2007-04-11 00:00:00.000000 UTC

Após geração do .csv foi tratado alguns dados como CPF, registro e número CNH.

### 3.3 MODELAGEM

Ao contrário do Modelo Estrela, será adotado a Flat Table, que consiste em uma tabela desnormalizada para o atingimento de objetivos específicos em ambientes de BI.

Sabemos que o Modelo Estrela é o mais amplamente utilizado, no entanto o modelo adequado depende de uma análise que envolve os pilares de tecnologia, segurança da informação e estratégia de negócios e objetivo.

emissao_cnh
CAR_INT_NR_CARTEIRA
SIT_INT_ID_SITE
ARQ_INT_NR_ARQUIVO
CAR_INT_CD_STATUS
CAR_STR_NR_RENACH
CAR_STR_NR_CPF
CAR_STR_DS_NOMELINHA1
CAR_STR_DS_NOMELINHA2
CAR_STR_DS_NOMEPAI
CAR_STR_NR_NOMEMAE
CAR_DAT_DT_NASCIMENTO
CAR_STR_ID_CATEGORIA
CAR_STR_ID_TIPO
CAR_STR_DS_UF
CAR_DAT_DT_EMISSAO
CAR_DAT_DT_PRIMEIRAHABILI...
CAR_DAT_DT_VALIDADE
CAR_DBL_NR_REGISTRO
CAR_STR_DS_DOADOR
CAR_STR_DS_OBSERVACAOLIN...
CAR_STR_DS_OBSERVACAOLIN...
CAR_STR_DS_OBSERVACAOLIN...
CAR_STR_DS_OBSERVACAOLIN...
CAR_STR_DS_OBSERVACAOLIN...
CAR_STR_DS_OBSERVACAOLIN...
CAR_STR_DS_OBSERVACAOLIN...
CAR_STR_DS_OBSERVACAOLIN...
CAR_INT_NR_HASHCNH
CAR_DBL_NR_TIPOGRAFICO
CAR_STR_CD_LOCALEMISSAO
CAR_STR_CD_OBSERVACOES
CAR_STR_NR_IDENTIDADE
CAR_INT_NR_LOTE
CAR_INT_NR_PEDIDO
CAR_STR_63CHARACTERS
CAR_DAT_DT_AUTORIZACAOD...
CAR_STR_DS_AUTORIZADORDE...
CAR_STR_CD_CIBETRA...

## CATÁLOGO DOS DADOS:

Catálogo de dados é um inventário organizado de ativos de dados na organização. Ele usa metadados para ajudar as organizações a gerenciarem seus dados. Também ajuda os profissionais de dados a coletar, organizar, acessar e enriquecer metadados para oferecer suporte à descoberta e governança de dados.

**Esquema atual**

ADD POLICY TAG

Filtro: Insira o nome ou o valor da propriedade

Nome do campo	Tipo	Modo	Compilação	Valor padrão	Tags de políticas	Descrição
SIT_INT_ID_SITE	STRING	NULLABLE		Valor padrão		Descrição Id do site

**Novos campos**

Editar como texto

1	Nome do campo *	CAR_INT_NR_CARTEIRA	Tipo *	INTEGER	Modo	NULLABLE	Descrição	Numero da carteira
2	Nome do campo *	ARO_INT_NR_ARQUIVO	Tipo *	INTEGER	Modo	NULLABLE	Descrição	Numero do arquivo
3	Nome do campo *	CAR_INT_CD_STATUS	Tipo *	INTEGER	Modo	NULLABLE	Descrição	Status carteira
4	Nome do campo *	CAR_STR_NR_RENACH	Tipo *	STRING	Modo	NULLABLE	Comprimir	11
5	Nome do campo *	CAR_STR_NR_CPF	Tipo *	STRING	Modo	NULLABLE	Comprimir	11
6	Nome do campo *	CAR_STR_DS_NOMELINHA1	Tipo *	STRING	Modo	NULLABLE	Comprimir	32
7	Nome do campo *	CAR_STR_DS_NOMELINHA2	Tipo *	STRING	Modo	NULLABLE	Comprimir	32
8	ADICIONAR CAMPO							

## LINHAGEM DOS DADOS:

A linhagem de dados é o processo de acompanhamento do fluxo de dados durante um período de tempo, fornecendo uma visão clara de onde os dados se originaram, como mudaram e do destino final dentro do pipeline de dados. Tentei utilizar o Dataplex que inclui uma funcionalidade muito útil para configurar e executar verificações de qualidade de dados em recursos, como tabelas do BigQuery e arquivos do Cloud Storage. Porém tive dificuldades e fiz um rascunho.

**Esquema atual**

ADD POLICY TAG

Filtro: Enter property name or value

Field name	Type	Mode	Key	Collation
SIT_INT_ID_SITE	STRING	NULLABLE	-	-
CAR_STR_NR_CPF	STRING(11)	NULLABLE	-	-
CAR_DAT_DT_NASCIMENTO	DATE	NULLABLE	-	-
CAR_DAT_DT_EMISSAO	DATE	NULLABLE	-	-
CAR_DAT_DT_PRIMEIRAHABILITACAO	DATE	NULLABLE	-	-
CAR_DAT_DT_VALIDADE	DATE	NULLABLE	-	-

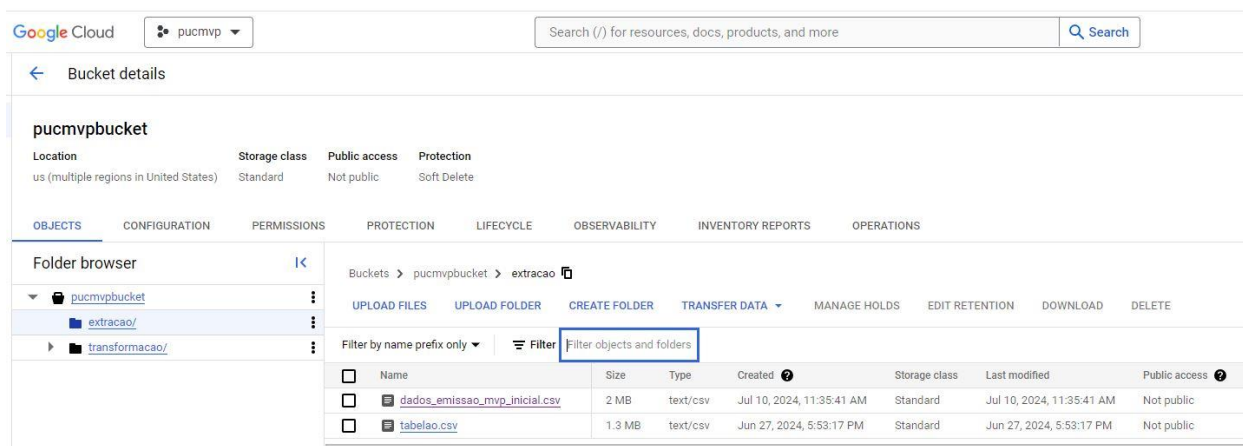
**Novos campos**

Editar como texto

1	Nome do campo *	CAR_INT_NR_CARTEIRA	Tipo *	INTEGER	Modo	NULLABLE	Descrição	Numero da carteira
2	Nome do campo *	ARO_INT_NR_ARQUIVO	Tipo *	INTEGER	Modo	NULLABLE	Descrição	Numero do arquivo
3	Nome do campo *	CAR_INT_CD_STATUS	Tipo *	INTEGER	Modo	NULLABLE	Descrição	Status carteira
4	Nome do campo *	CAR_STR_NR_RENACH	Tipo *	STRING	Modo	NULLABLE	Comprimir	11
5	Nome do campo *	CAR_STR_NR_CPF	Tipo *	STRING	Modo	NULLABLE	Comprimir	11
6	Nome do campo *	CAR_STR_DS_NOMELINHA1	Tipo *	STRING	Modo	NULLABLE	Comprimir	32
7	Nome do campo *	CAR_STR_DS_NOMELINHA2	Tipo *	STRING	Modo	NULLABLE	Comprimir	32
8	ADICIONAR CAMPO							

## 4. CARGA

A etapa de carga do dataset para Cloud Storage / Bucket, foi feita manualmente e será utilizada e atualizada pela pipeline desenvolvida utilizando Apache Spark.



Google Cloud pucmvp Search (/) for resources, docs, products, and more

Bucket details

pucmvpbucket

Location: us (multiple regions in United States) Storage class: Standard Public access: Not public Protection: Soft Delete

OBJECTS CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE OBSERVABILITY INVENTORY REPORTS OPERATIONS

Folder browser

Buckets > pucmvpbucket > extracao

UPLOAD FILES UPLOAD FOLDER CREATE FOLDER TRANSFER DATA MANAGE HOLDS EDIT RETENTION DOWNLOAD DELETE

Filter by name prefix only Filter objects and folders

Name	Size	Type	Created	Storage class	Last modified	Public access
dados_emissao_mvp_inicial.csv	2 MB	text/csv	Jul 10, 2024, 11:35:41 AM	Standard	Jul 10, 2024, 11:35:41 AM	Not public
tabelao.csv	1.3 MB	text/csv	Jun 27, 2024, 5:53:17 PM	Standard	Jun 27, 2024, 5:53:17 PM	Not public

Pipeline Executada

[https://github.com/leonardosva/mvp\\_cnh\\_eng\\_dados/blob/main/pipeline.pdf](https://github.com/leonardosva/mvp_cnh_eng_dados/blob/main/pipeline.pdf)



## 5. ANALISE

### QUALIDADE DOS DADOS:

Os fatores qualitativos podem variar de caso para caso, mas de forma geral um dado de qualidade apresenta **completude**, **conformidade**, **precisão**, **consistência** e **integridade** (possui fontes e processos confiáveis e rastreáveis).

No conjunto inicial, os tipos de dados já apresentam padrões aceitáveis, mas para as respostas iniciais, devido uma particularidade, precisamos os campos **data** para **timestamp**, **string** para **integer**.

Alteramos o datatype pela pipeline de algumas colunas do DataSet para melhor representar as respostas iniciais.

SCHEMA

DETAILS

PREVIEW

LINEAGE

DATA PROFILE

DATA QUALI

Filter

Enter property name or value

<input type="checkbox"/>	Field name	Type	Mode	Key	Colla
<input type="checkbox"/>	SIT_INT_ID_SITE	STRING	NULLABLE	-	-
<input type="checkbox"/>	CAR_INT_NR_CARTEIRA	INTEGER	NULLABLE	-	-
<input type="checkbox"/>	ARQ_INT_NR_ARQUIVO	INTEGER	NULLABLE	-	-
<input type="checkbox"/>	CAR_INT_CD_STATUS	INTEGER	NULLABLE	-	-
<input type="checkbox"/>	CAR_STR_NR_RENACH	STRING(11)	NULLABLE	-	-
<input type="checkbox"/>	CAR_STR_NR_CPF	STRINGS(11)	NULLABLE	-	-
<input type="checkbox"/>	CAR_STR_DS_NOMELINHA1	STRING(32)	NULLABLE	-	-
<input type="checkbox"/>	CAR_STR_DS_NOMELINHA2	STRING(32)	NULLABLE	-	-
<input type="checkbox"/>	CAR_DAT_DT_NASCIMENTO	DATE	NULLABLE	-	-
<input type="checkbox"/>	CAR_STR_ID_CATEGORIA	INTEGER	NULLABLE	-	-
<input type="checkbox"/>	CAR_STR_ID_TIPO	STRING(1)	NULLABLE	-	-
<input type="checkbox"/>	CAR_STR_DS_UF	STRING(2)	NULLABLE	-	-
<input type="checkbox"/>	CAR_DAT_DT_EMISSAO	DATE	NULLABLE	-	-
<input type="checkbox"/>	CAR_DAT_DT_PRIMEIRAHABILITACAO	DATE	NULLABLE	-	-

Filter

Enter property name or value

<input type="checkbox"/>	Field name	Type	Mode	Key	Collat
<input type="checkbox"/>	SIT_INT_ID_SITE	INTEGER	NULLABLE	-	-
<input type="checkbox"/>	CAR_INT_CD_STATUS	INTEGER	NULLABLE	-	-
<input type="checkbox"/>	CAR_STR_NR_CPF	INTEGER	NULLABLE	-	-
<input type="checkbox"/>	CAR_STR_DS_NOMELINHA1	STRING	NULLABLE	-	-
<input type="checkbox"/>	CAR_DAT_DT_NASCIMENTO	TIMESTAMP	NULLABLE	-	-
<input type="checkbox"/>	CAR_STR_ID_CATEGORIA	STRING	NULLABLE	-	-
<input type="checkbox"/>	CAR_STR_ID_TIPO	STRING	NULLABLE	-	-
<input type="checkbox"/>	CAR_STR_DS_UF	STRING	NULLABLE	-	-
<input type="checkbox"/>	CAR_DAT_DT_EMISSAO	TIMESTAMP	NULLABLE	-	-
<input type="checkbox"/>	CAR_DAT_DT_PRIMEIRAHABILITACAO	TIMESTAMP	NULLABLE	-	-
<input type="checkbox"/>	CAR_DAT_DT_VALIDADE	TIMESTAMP	NULLABLE	-	-
<input type="checkbox"/>	CAR_DBL_NR_REGISTRO	FLOAT	NULLABLE	-	-
<input type="checkbox"/>	CAR_STR_DS_OBSERVACAOLINHA1	STRING	NULLABLE	-	-
<input type="checkbox"/>	CAR STR DS OBSERVACAOLINHA2	STRING	NULLABLE	-	-

### SOLUÇÃO DO PROBLEMA:

Respostas

1- Total Emissão por UF

```
In [12]: # Emissao de Ct's por UF
dframe_filtrado_1 = dframe_filtrado.groupby(["DS_UF"]).agg(F.count("NR_CPF").alias("QTD_CNH"))
dframe_filtrado_1.show()
```

[Stage 21:> (0 + 1) / 1]

DS_UF	QTD_CNH
GO	1683
RJ	5361
MA	331
DF	424

2- Porcentagem % Emissão por UF.

```
In [13]: # Count qtd Linhas Dataframe
count_c1 = dframe_filtrado.count()

# Porcentagem de Emissões por UF
dframe_filtrado_2 = dframe_filtrado.groupBy(["DS_UF"]).count().withColumn('% UF', func.round((func.col('count')/count_c1)*100,2))
.orderBy('count', ascending=False) \
.show()
```

[Stage 36:> (0 + 1) / 1]

DS_UF	count	% UF
RJ	5361	68.74
GO	1683	21.58
DF	424	5.44
MA	331	4.24

### 3- Porcentagem % de emissão por TIPO CNH. (Permissionário x Condutor).

```
In [14]: # Count qtd linhas Dataframe
count_c1 = dframe_filtrado.count()

# Porcentagem de Emissões por Tipo CNH, onde : C = Condutor e P = Permissionário
dframe_filtrado_3 = dframe_filtrado.groupBy(["DS_TIPO"]).count().withColumn('% Tipo CNH', func.round((func.col('count') / count_c1) * 100, 2)
.orderBy('count', ascending=False) \
.show()
```

DS_TIPO	count	% Tipo CNH
C	6733	86.33
P	1066	13.67

### 4- Media tipo categoria por idade.

> > Não respondida, pois na Pipeline não consegui tratar literal Idade.

### 5- Media idade primeira habilitação.

>> Não respondida, pois na Pipeline não consegui tratar literal Idade.

```
In [14]: # Tentativa de tratamento Cálculo IDADE para responder perguntas 4 e 5 .
# Consegui forçar apenas passando a String lit("1982-08-06"), mas ai todos ficavam com 41 anos.
# Não consegui buscar do dataframe a DT_Nascimento para cada registro.

from pyspark.sql.functions import to_timestamp
from pyspark.sql.functions import to_date, datediff, floor, current_date, lit, concat_ws, col

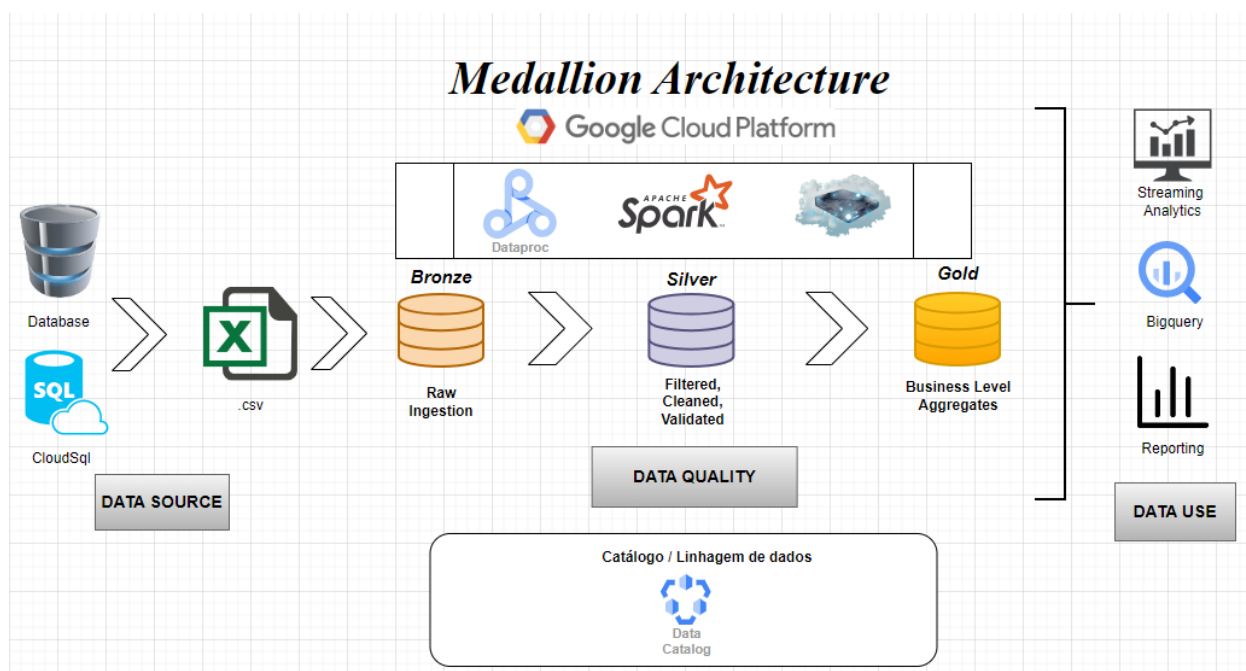
age_column = floor(datediff(current_date(), to_date(concat_ws("-", lit("1982-08-06"), "yyyy-MM-dd")), "yyyy-MM-dd") / 365)
#age_column = floor(datediff(current_date(), to_date(concat_ws("-", lit(dframe_filtrado.select("DT_NASCIMENTO")), "yyyy-MM-dd"), "yyyy-MM-dd"))
#print(age_column)

# Add the new "Idade" column to the DataFrame
dframe_filtrado = dframe_filtrado.withColumn("Idade", age_column.cast("int"))
dframe_filtrado_idade = dframe_filtrado.select("Idade")
dframe_filtrado_idade.show(2)
```

Idade
41
41

only showing top 2 rows

## 6. ARQUITETURA



A arquitetura do medalhão exemplifica uma série de camadas de dados que denotam a qualidade dos dados armazenados. Recomendasse adotar uma abordagem multicamadas para criar uma única fonte de verdade para produtos de dados. Essa arquitetura garante atomicidade, consistência, isolamento e durabilidade à medida que os dados passam por várias camadas de validações e transformações antes de serem armazenados em uma disposição otimizada para análise eficiente. Os termos bronze (bruto), prata (validado) e ouro (enriquecido) descrevem a qualidade dos dados em cada uma dessas camadas.

## 7. FLUXO PIPELINE

## Importação e Definição Project na GCP

```
In [2]: import os
import requests
import tempfile
import pyspark.sql.functions as F
from pyspark.sql import SparkSession
from pyspark.ml.feature import Bucketizer
from pyspark.sql.functions import col
from pyspark.sql.window import Window
from google.cloud import storage
from google.cloud import bigquery
import numpy as np

In [3]: # Definição do projeto na GCP
project_id = "pucmvp"

# Definindo a sessão spark
spark = SparkSession.builder.appName("AnaliseCNH").getOrCreate()
```

## Camada Bronze:

1. Raw Ingestion - Camada Bronze / Upload do arquivo realizado manualmente para o Bucket

In [5]: #Lendo o dataset do Cloud Storage

```
#dframe = spark.read.csv("gs://pucmvpbucket/extracao/tabelao.csv",header=True, inferSchema=True, sep=';')
|
dframe = spark.read.csv("gs://pucmvpbucket/extracao/dados_emissao_mvp_inicial.csv",header=True, inferSchema=True, sep=',')
```

In [6]: # Printando schema do Dataset

```
dframe.printSchema()
```

```
root
 |-- SIT_INT_ID_SITE: integer (nullable = true)
 |-- CAR_INT_CD_STATUS: integer (nullable = true)
 |-- CAR_STR_NR_CPF: long (nullable = true)
 |-- CAR_STR_DS_NOMEINHA1: string (nullable = true)
 |-- CAR_DAT_DT_NASCIMENTO: string (nullable = true)
 |-- CAR_STR_ID_CATEGORIA: string (nullable = true)
 |-- CAR_STR_ID_TIPO: string (nullable = true)
 |-- CAR_STR_DS_UF: string (nullable = true)
 |-- CAR_DAT_DT_EMISSAO: string (nullable = true)
 |-- CAR_DAT_DT_PRIMEIRAHABILITACAO: string (nullable = true)
 |-- CAR_DAT_DT_VALIDADE: string (nullable = true)
 |-- CAR_DBL_NR_REGISTRO: double (nullable = true)
 |-- CAR_STR_DS_OBSERVACAOLINHA1: string (nullable = true)
 |-- CAR_STR_DS_OBSERVACAOLINHA2: string (nullable = true)
```

In [7]: # Imprimindo 2 Linhas do Dataset Carregado

```
dframe.show(2)
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
|SIT_INT_ID_SITE|CAR_INT_CD_STATUS|CAR_STR_NR_CPF|CAR_STR_DS_NOMEINHA1|CAR_DAT_DT_NASCIMENTO|CAR_STR_ID_CATEGORIA|CAR_STR_ID_T|
IPO|CAR_STR_DS_UF|CAR_DAT_DT_EMISSAO|CAR_DAT_DT_PRIMEIRAHABILITACAO|CAR_DAT_DT_VALIDADE|CAR_DBL_NR_REGISTRO|CAR_STR_DS_OBSER|
VACAOLINHA1|CAR_STR_DS_OBSERVACAOLINHA2|
+-----+-----+-----+-----+-----+-----+-----+-----+
|          8|          22| 44815564934| ANTONILSON FRAZAO...| 1966-11-24 00:00:...|          B |
C|          MA|2024-07-08 00:00:...|          2007-07-10 00:00:...|2029-06-24 00:00:...| 4.13796069E9|
D|          D|          D|          D|          D|          D|          D|          D|
+-----+-----+-----+-----+-----+-----+-----+-----+
|          8|          22| 61083660102| LUCIANO SILVA DOS...| 1999-04-23 00:00:...|          B |
P|          MA|2024-07-08 00:00:...|          2024-07-08 00:00:...|2025-07-07 00:00:...| 8.684790978E9|
D|          E|          E|          E|          E|          E|          E|          E|
+-----+-----+-----+-----+-----+-----+-----+-----+
only showing top 2 rows
```

In [8]: # Quantidade de Registros e Colunas do Dataset

```
print((dframe.count(), len(dframe.columns)))
```

```
[Stage 3:>                                (0 + 1) / 1]
```

```
(10000, 14)
```

## Camada Prata:

2. Filtered, Cleaned, Validated - Camada Silver

```
In [9]: # Remover Linhas/dados CPF duplicados
dframe = dframe.dropDuplicates().na.fill(2)
```

```
In [10]: # Quantidade de Registros e Colunas do Dataset após apagar Linhas/registros CPF Duplicados
print((dframe.count(), len(dframe.columns)))
```

```
[Stage 6:>                                (0 + 1) / 1]
(7799, 14)
```

```
In [11]: from pyspark.sql.functions import col, count

# Mapeamento dos dados:
# Conversão de datatype >> dframe.withColumn("CAR_DBL_NR_REGISTRO", col("CAR_DBL_NR_REGISTRO").cast("integer"))
# Conversão de nomenclatura >> dframe.withColumnRenamed("CAR_INT_CD_STATUS", "CD_STATUS")
# Converter tipo de coluna double para integer / string para integer e datas para Timestamp

dframe_filtrado = \
dframe.withColumn("CAR_DBL_NR_REGISTRO", col("CAR_DBL_NR_REGISTRO").cast("integer")) \
.withColumnRenamed("CAR_DBL_NR_REGISTRO", "NR_REGISTRO") \
.withColumnRenamed("SIT_INT_ID_SITE", "ID_SITE") \
.withColumnRenamed("CAR_INT_CD_STATUS", "CD_STATUS") \
.withColumn("CAR_STR_NR_CPF", col("CAR_STR_NR_CPF").cast("integer")) \
.withColumnRenamed("CAR_STR_NR_CPF", "NR_CPF") \
.withColumnRenamed("CAR_STR_DS_NOMELINHA1", "DS_NOMELINHA1") \
.withColumn("CAR_DAT_DT_NASCIMENTO", col("CAR_DAT_DT_NASCIMENTO").cast("timestamp")) \
.withColumnRenamed("CAR_DAT_DT_NASCIMENTO", "DT_NASCIMENTO") \
.withColumnRenamed("CAR_STR_ID_CATEGORIA", "DS_CATEGORIA") \
.withColumnRenamed("CAR_STR_ID_TIPO", "DS_TIPO") \
.withColumnRenamed("CAR_STR_DS_UF", "DS_UF") \
.withColumn("CAR_DAT_DT_EMISSAO", col("CAR_DAT_DT_EMISSAO").cast("timestamp")) \
.withColumnRenamed("CAR_DAT_DT_EMISSAO", "DT_EMISSAO") \
.withColumn("CAR_DAT_DT_PRIMEIRAHABILITACAO", col("CAR_DAT_DT_PRIMEIRAHABILITACAO").cast("timestamp")) \
.withColumnRenamed("CAR_DAT_DT_PRIMEIRAHABILITACAO", "DT_PRIMEIRAHABILITACAO") \
.withColumn("CAR_DAT_DT_VALIDADE", col("CAR_DAT_DT_VALIDADE").cast("timestamp")) \
.withColumnRenamed("CAR_DAT_DT_VALIDADE", "DT_VALIDADE") \
.withColumnRenamed("CAR_STR_DS_OBSERVACAOLINHA1", "DS_OBSERVACAOLINHA1") \
.withColumnRenamed("CAR_STR_DS_OBSERVACAOLINHA2", "DS_OBSERVACAOLINHA2")

# Printando Dataset com tipo e nome alterados
dframe_filtrado.printSchema()
```

```
root
|-- ID_SITE: integer (nullable = true)
|-- CD_STATUS: integer (nullable = true)
|-- NR_CPF: integer (nullable = true)
|-- DS_NOMELINHA1: string (nullable = true)
|-- DT_NASCIMENTO: timestamp (nullable = true)
|-- DS_CATEGORIA: string (nullable = true)
|-- DS_TIPO: string (nullable = true)
|-- DS_UF: string (nullable = true)
|-- DT_EMISSAO: timestamp (nullable = true)
|-- DT_PRIMEIRAHABILITACAO: timestamp (nullable = true)
|-- DT_VALIDADE: timestamp (nullable = true)
|-- NR_REGISTRO: integer (nullable = true)
|-- DS_OBSERVACAOLINHA1: string (nullable = true)
|-- DS_OBSERVACAOLINHA2: string (nullable = true)
```

```
In [13]: # Gerando datagrid sem coluna acima deletada NR_REGISTRO
dframe_filtrado.show(2)

# Quantidade de Registros e Colunas do Dataset após deletar coluna NR_REGISTRO
print((dframe_filtrado.count(), len(dframe_filtrado.columns)))
```

ID_SITE	CD_STATUS	NR_CPF	DS_NOMEINHA1	DT_NASCIMENTO	DS_CATEGORIA	DS_TIPO	DS_UF	DT_EMISSAO	DT_PRIMEIRA
RAHABILITACAO		DT_VALIDADE	DS_OBSERVACAOINHA1	DS_OBSERVACAOINHA2					
9	22	1038201064	VITORIA MARIA DE ...	2002-06-03 00:00:00	B	C	DF	2024-07-08 00:00:00	2022-0
4-29 00:00:00	2031-04-28 00:00:00	A	...	EAR	...				
8	22	147297167	AIRTON KOCHHANN ...	1969-10-12 00:00:00	AE	C	MA	2024-07-09 00:00:00	1992-0
3-18 00:00:00	2029-07-08 00:00:00		null	EAR					

only showing top 2 rows

```
[Stage 15:> (0 + 1) / 1]
(7799, 13)
```

```
In [12]: # Eliminando coluna/dado Irrelevante para analise NR_REGISTRO
dframe_filtrado = dframe_filtrado.drop("NR_REGISTRO")

# Printando schema sem o registro acima deletado NR_REGISTRO
dframe_filtrado.printSchema()
```

```
root
 |-- ID_SITE: integer (nullable = true)
 |-- CD_STATUS: integer (nullable = true)
 |-- NR_CPF: integer (nullable = true)
 |-- DS_NOMEINHA1: string (nullable = true)
 |-- DT_NASCIMENTO: timestamp (nullable = true)
 |-- DS_CATEGORIA: string (nullable = true)
 |-- DS_TIPO: string (nullable = true)
 |-- DS_UF: string (nullable = true)
 |-- DT_EMISSAO: timestamp (nullable = true)
 |-- DT_PRIMEIRA: timestamp (nullable = true)
 |-- DT_VALIDADE: timestamp (nullable = true)
 |-- DS_OBSERVACAOINHA1: string (nullable = true)
 |-- DS_OBSERVACAOINHA2: string (nullable = true)
```

```
In [14]: # Tentativa de tratamento Cálculo IDADE para responder perguntas 4 e 5 .
# Consegui forçar apenas passando a String lit("1982-08-06")), mas ai todos ficavam com 41 anos.
# Não consegui buscar do dataframe a DT_Nascimento para cada registro.

from pyspark.sql.functions import to_timestamp
from pyspark.sql.functions import to_date, datediff, floor, current_date, lit, concat_ws, col

age_column = floor(datediff(current_date(), to_date(concat_ws("-", lit("1982-08-06")), "yyyy-MM-dd")) / 365)
#age_column = floor(datediff(current_date(), to_date(concat_ws("-", lit(dframe_filtrado.select("DT_NASCIMENTO"))), "yyyy-MM-dd"),
#print(age_column)

# Add the new "Idade" column to the DataFrame
dframe_filtrado = dframe_filtrado.withColumn("Idade", age_column.cast("int"))
dframe_filtrado_idade = dframe_filtrado.select("Idade")
dframe_filtrado_idade.show(2)
```

Idade
41
41

only showing top 2 rows

```
In [15]: # Armazenar a transformação no bucket Silver
# Defina o caminho para o arquivo Parquet no bucket
path_parquet = "gs://pucmpbucket/transformacao/analise_tratada_3.parquet"

# Salve as novas colunas em Parquet no Cloud Storage
dframe_filtrado.write.format("parquet").option("path", path_parquet).save()
```

Imagem abaixo evidência



Folder browser

- pucmvpbucket
  - extracao/
  - transformacao/
    - analise\_tratada\_2.parquet/
    - analise\_tratada\_3.parquet/
    - analise\_tratada.parquet/
    - analise\_tratada1.parquet/
    - AnaliseCNH\_Tratado.parquet/

Buckets > pucmvpbucket > transformacao

[UPLOAD FILES](#)
[UPLOAD FOLDER](#)
[CREATE FOLDER](#)
[TRANSFER DATA](#)

[DOWNLOAD](#)
[DELETE](#)

Filter by name prefix only Filter Filter objects and folders

<input type="checkbox"/>	Name	Size	Type	Created
<input type="checkbox"/>	AnaliseCNH_Tratado.parquet/	—	Folder	—
<input type="checkbox"/>	analise_tratada.parquet/	—	Folder	—
<input type="checkbox"/>	analise_tratada1.parquet/	—	Folder	—
<input type="checkbox"/>	analise_tratada_2.parquet/	—	Folder	—
<input type="checkbox"/>	analise_tratada_3.parquet/	—	Folder	—

## Camada Ouro:

### 3. Carregamento de Dados / Business Level Aggregates - Camada Ouro

```
In [16]: # Autenticação no BigQuery
cliente_bq = bigquery.Client()

# Defina o nome do projeto e do conjunto de dados no BigQuery
dataset_id = "emissao_raw"
```

```
In [ ]: from pyspark.sql import SparkSession
# Importando tabela no BigQuery
dframe_filtrado.write.format("bigquery").option("temporaryGcsBucket", "").option("writeMethod", "DIRECT").option("project", proje
```

```
In [17]: # Emissao de Ct's por UF
dframe_filtrado_1 = dframe_filtrado.groupBy(["DS_UF"]).agg(F.count("NR_CPF").alias("QTD_CNH"))
dframe_filtrado_1.show()
```

```
+-----+-----+
|DS_UF|QTD_CNH|
+-----+-----+
| GO | 1683 |
| RJ | 5361 |
| MA | 331 |
| DF | 424 |
+-----+-----+
```

```
In [19]: # Count qtd Linhas Dataframe
count_c1 = dframe_filtrado.count()

# Porcentagem de Emissões por Tipo CNH, onde : C = Condutor e P = Permissionário
dframe_filtrado_3 = dframe_filtrado.groupBy(["DS_TIPO"]).count().withColumn('% Tipo CNH', func.round((func.col('count')/count_c1)
.orderBy('count', ascending=False) \
.show()
```

[Stage 54:] (0 + 1) / 1]

```
+-----+-----+
|DS_TIPO|count|% Tipo CNH|
+-----+-----+
| C | 6733 | 86.33 |
| P | 1066 | 13.67 |
+-----+-----+
```

```
In [20]: # Count qtd Linhas Dataframe
count_c1 = dframe_filtrado.count()

# Porcentagem de Emissões por UF
dframe_filtrado_2 = dframe_filtrado.groupBy(["DS_UF"]).count().withColumn('% UF', func.round((func.col('count')/count_c1)*100,2)
.orderBy('count', ascending=False) \
.show()
```

```
+-----+-----+
|DS_UF|count| % UF |
+-----+-----+
| RJ | 5361 |68.74 |
| GO | 1683 |21.58 |
| DF | 424 | 5.44 |
| MA | 331 | 4.24 |
+-----+-----+
```

**Imagem abaixo evidência**

←

Bucket details

OBJECTS

CONFIGURATION

PERMISSIONS

PROTECTION

LIFECYCLE

OBSERVABILITY

INVENTORY REPORTS

Folder browser

▼

📁 pucmvpbucket

⋮

📁 entrega/

⋮

📁 extracao/

⋮

▶ 📁 transformacao/

⋮

Buckets > pucmvpbucket > entrega 📁

UPLOAD FILES

UPLOAD FOLDER

CREATE FOLDER

TRANSFER DATA ▼

MANAGE HOL

DELETE

Filter by name prefix only ▼

☰ Filter

Filter objects and folders

<input type="checkbox"/>	Name	Size	Type	Created ?
<input type="checkbox"/>	📄 dados_emissao_mvp_final.csv	2 MB	text/csv	Jul 12, 2024, 12:40:33 AM

## 8. AUTO AVALIAÇÃO

A ideia do projeto como um todo, é demonstrar uma abordagem bem estruturada e abrangente para o desenvolvimento de um pipeline tratamento/análise de dados de emissão de CNH. O projeto utiliza efetivamente tecnologias de nuvem, abordando aspectos de qualidade de dados e descrevendo um modelo claro para execução. As explicações e justificativas detalhadas para cada etapa indicam uma compreensão completa dos requisitos e desafios do projeto.

Inicialmente, fiz uma busca nos sites sugeridos para identificar dataset's gratuitos, porem fiquei mais confortável em manipular e trabalhar com dados do meu dia-a-dia (emissão de carteiras de habilitação), pois as perguntas já eram bem definidas em minha cabeça.

Posteriormente, busquei diversos vídeos e tutoriais que descreviam e exemplificavam o que é e como criar um modelo de datawarehouse / datalake utilizando a linguagem Python e ferramentas web que facilitariam o desenvolvimento de uma pipeline.

Cheguei nas ferramentas (Python, Dataproc, Dataplex, Nifi, BigQuery, Cloud Storage, Metabase, Datalog, Dataflow, etc .), e a sugerida pelos professores [ DataBricks ].

Tive bastante dificuldade em colocar na prática o fluxo para cada etapa: arquitetura, modelo e de análise dos dados, escolha do melhor dado, tratamento do dado em especifico e com isso minhas perguntas ficaram mais difíceis de responder.

Optei pela Flat Table na criação do modelo, após muita dificuldade em criar as tabelas fato e dimensão. Nesse momento a ideia começou a ficar mais clara para o desenvolvimento (os encontros e o discord, tambem ajudaram muito a clarear as ideias).

No contexto geral, fiquei muito satisfeito com as pesquisas e a capacidade de colocar em prática o aprendizado. Utilizei:

**Cloud Storage** - Para criar o bucket e as estruturas de arquivos extração, transformação e entrega.

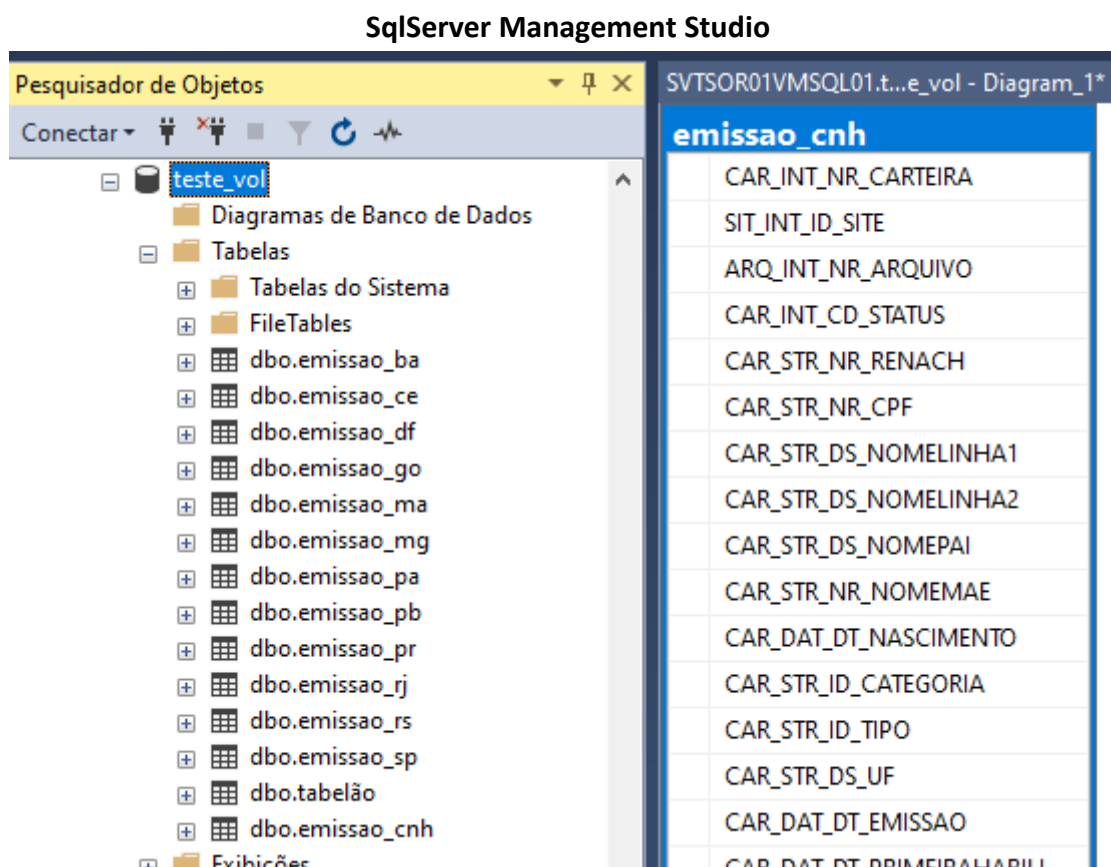
**DataProc** - Para cria o cluster e usar a web interface do júpiter para desenvolver a pipeline.

**BigQuery** - Para criar a estruturas de tabelas, catálogo de dados, carga e consultas dos dados.


**DataPlex** - Para tentar criar a linhagem dos dados.


**Metabase** - Para ler o dataset no BigQuery e gerar dashboards das respostas.

## Algumas Imagens / Evidências das Ferramentas Utilizadas para desenvolvimento do MVP





## Cloud Storage Image




pucmvp

Search (/) for resources, docs, products, and more

 Search


← Bucket details
GO TO PATH

Back to parent page

OBJECTS

CONFIGURATION

PERMISSIONS


PROTECTION


LIFECYCLE

OBSERVABILITY

INVENTORY REPORTS

OPERATIONS



Folder browser


pucmvpbucket

entrega/

extracao/

transformacao/


Name


Size

Type

Created ?

Storage class

Last modified


entrega/


—

Folder

—

—

—


extracao/


—

Folder

—

—

—


transformacao/


—


Folder

—


—


—




pucmvp

Search (/) for resources, docs, products, and more

 Search


← Bucket details
GO TO PATH

OBJECTS

CONFIGURATION

PERMISSIONS


PROTECTION


LIFECYCLE

OBSERVABILITY

INVENTORY REPORTS

OPERATIONS



Folder browser


pucmvpbucket

entrega/

extracao/

transformacao/



Name

Size

Type

Created ?

Storage class



dados\_emissao\_mv...

2 MB

text/csv

Jul 10, 2024, 11:35:41 AM

Standard



tabelao.csv


1.3 MB

text/csv


Jun 27, 2024, 5:53:17 PM


Standard




pucmvp

Search (/) for resources, docs, products, and more

 Search


Bucket details
GO TO PATH

OBJECTS

CONFIGURATION

PERMISSIONS


PROTECTION


LIFECYCLE

OBSERVABILITY

INVENTORY REPORTS

OPERATIONS



Folder browser


pucmvpbucket

entrega/

extracao/

transformacao/


Name


Size

Type

Created ?

Storage class

Last modified


AnaliseCNH\_Tratado.parquet/


—

Folder

—

—

—


analise\_tratada.parquet/


—

Folder

—

—

—


analise\_tratada1.parquet/


—

Folder

—

—

—


analise\_tratada2.parquet/


—

Folder

—

—

—


analise\_tratada3.parquet/

—

Folder

—

—

—

## Dataprocc Cluster Image

Google Cloud pucmvp Search (/) for resources, docs, products, and more

Navigation menu clusters + CREATE CLUSTER REFRESH START STOP DELETE REGIONS + 5 RECOMMENDED

Filter Search cluster by properties, press Enter

Name	Status	Region	Zone	Total worker nodes	Flexible VMs?	Scheduled deletion
mvp	Running	us-central1	us-central1-c	0	No	Off

Google Cloud pucmvp Search (/) for resources, docs, products, and more Search

Cluster details SUBMIT JOB REFRESH START STOP DELETE VIEW LOGS

Name mvp  
Cluster UUID 0bbfc241-80aa-4fa0-aecb-d3e9b42ea1cd  
Type Dataprocc Cluster  
Status Running

MONITORING JOBS VM INSTANCES CONFIGURATION WEB INTERFACES

Google Cloud pucmvp Search (/) for resources, docs, products, and more Search

Cluster details SUBMIT JOB REFRESH START STOP DELETE VIEW LOGS

Provides access to the web interfaces of default and selected optional components on the cluster. [Learn more](#)

- [YARN ResourceManager](#)
- [MapReduce Job History](#)
- [Spark History Server](#)
- [HDFS NameNode](#)
- [YARN Application Timeline](#)
- [HiveServer2 \(mvp-m\)](#)
- [Tez](#)
- [Jupyter](#)
- [JupyterLab](#)

pucmvp > mvp

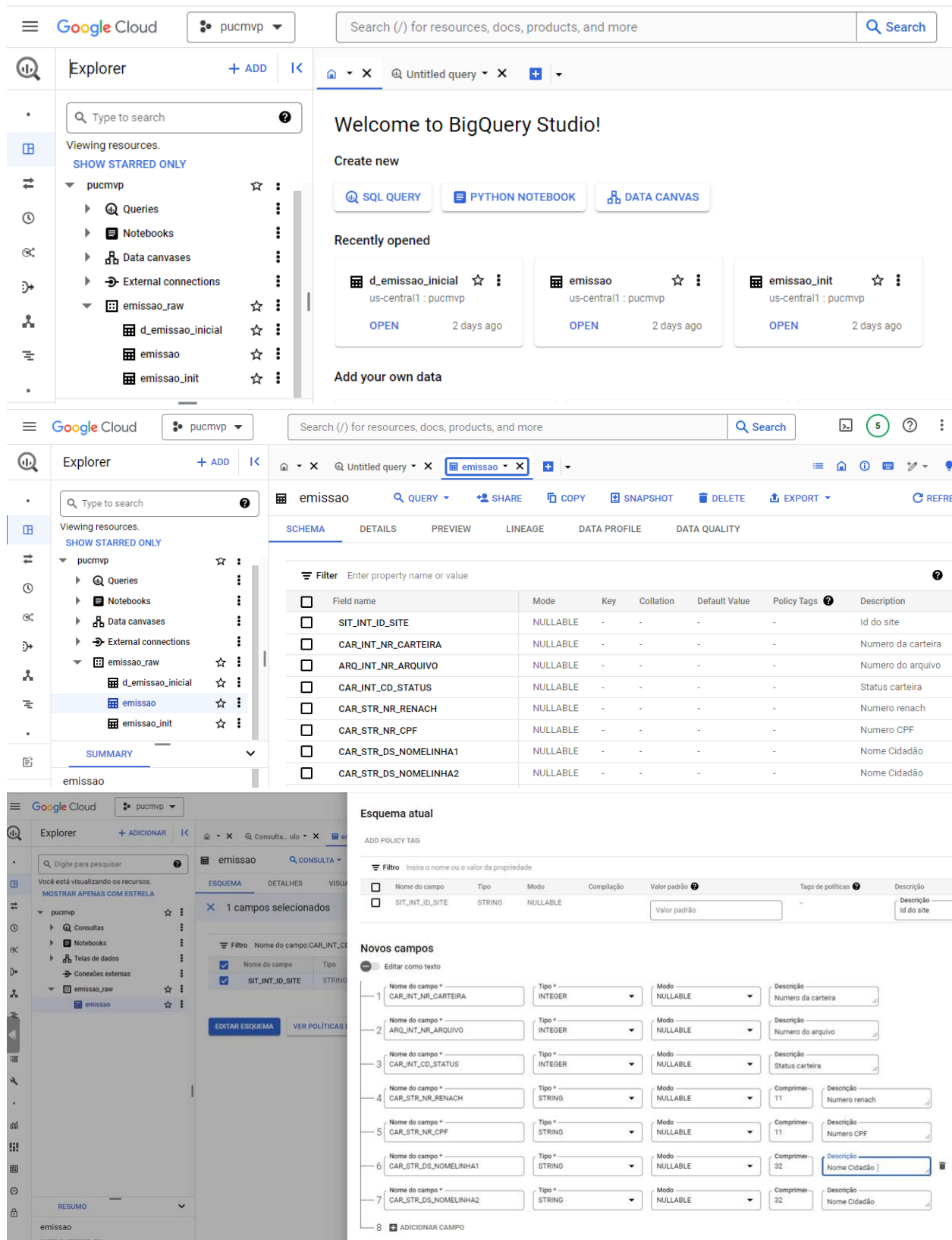
Jupyter pipeline Last Checkpoint: uma hora atrás (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Code nbdiff

```
In [3]: import os
import requests
import tempfile
import pyspark.sql.functions as F
import pyspark.sql.functions as func
from pyspark.sql.types import *
from pyspark.sql import SparkSession
from pyspark.sql import Row
from pyspark.ml.feature import Bucketizer
from pyspark.sql.functions import col
from pyspark.sql.window import window
from google.cloud import storage
from google.cloud import bigquery
import numpy as np
```

## BigQuery Image



The screenshot displays the Google Cloud BigQuery Studio interface. The top navigation bar includes the Google Cloud logo, a project selector set to 'pucmpv', and a search bar. The left sidebar shows the Explorer view with a tree structure of resources: Queries, Notebooks, Data canvases, External connections, and a folder named 'emissao\_raw' containing 'd\_emissao\_inicial', 'emissao', and 'emissao\_init'. The main panel shows the 'Welcome to BigQuery Studio!' message with options to 'Create new' (SQL QUERY, PYTHON NOTEBOOK, DATA CANVAS) and 'Recently opened' queries. Below this, the 'emissao' table is selected, and the 'SCHEMA' tab is active, displaying a table of fields with their names, modes, keys, collations, default values, policy tags, and descriptions.

Field name	Mode	Key	Collation	Default Value	Policy Tags	Description
SIT_INT_ID_SITE	NULLABLE	-	-	-	-	Id do site
CAR_INT_NR_CARTEIRA	NULLABLE	-	-	-	-	Numero da carteira
ARQ_INT_NR_ARQUIVO	NULLABLE	-	-	-	-	Numero do arquivo
CAR_INT_CD_STATUS	NULLABLE	-	-	-	-	Status carteira
CAR_STR_NR_RENACH	NULLABLE	-	-	-	-	Numero renach
CAR_STR_NR_CPF	NULLABLE	-	-	-	-	Numero CPF
CAR_STR_DS_NOMELINHA1	NULLABLE	-	-	-	-	Nome Cidadão
CAR_STR_DS_NOMELINHA2	NULLABLE	-	-	-	-	Nome Cidadão

Below the schema view, there is a section for 'Esquema atual' (Current Schema) and 'Novos campos' (New Fields). The 'Novos campos' section allows for adding new fields with a form that includes fields for 'Nome do campo' (Field Name), 'Tipo' (Type), 'Modo' (Mode), 'Compressão' (Compression), and 'Descrição' (Description). The form is currently empty, and the 'ADICIONAR CAMPO' (Add Field) button is visible at the bottom.

Google Cloud

pucmvp

Search (/) for res

Explorer

+ ADD

I<

Type to search

?

Viewing resources.

SHOW STARRED ONLY

pucmvp

Queries

Notebooks

Data canvases

External connections

emissao\_raw

d\_emissao\_inicial

emissao

emissao\_init

SUMMARY

d\_emissao\_inicial

pucmvp.emissao\_raw

Last modified: Jul 10, 2024, 11:30:20 AM UTC-3

Data location: us-central1

Description

Untitled query

d\_emiss... ial

+

d\_emissao\_inicial

QUERY

SHARE

SCHEMA

DETAILS

PREVIEW

LINEAGE

Table info

Table ID: pucmvp.emissao\_raw.d\_emissao\_inicial

Created: Jul 10, 2024, 11:30:20 AM UTC-3

Last modified: Jul 10, 2024, 11:30:20 AM UTC-3

Table expiration: Jul 30, 2024, 11:30:20 AM UTC-3

Data location: us-central1

Default collation

Default rounding mode: ROUNDING\_MODE\_UNSPECIFIED

Case insensitive: false

Description

Labels

Primary key(s)

Tags

Storage info

Number of rows: 10,000

Total logical bytes: 1.21 MB

Active logical bytes: 1.21 MB

Long term logical bytes: 0 B

Total physical bytes: 260.08 KB

Active physical bytes: 260.08 KB

Long term physical bytes: 0 B

Time travel physical: 0 B

Google Cloud

pucmvp

Search (/) for resources, docs, p

Untitled query

d\_emiss... ial

+

d\_emissao\_inicial

QUERY

SHARE

COPY

SNAPSHOT

DELETE

SCHEMA

DETAILS

PREVIEW

LINEAGE

DATA PROFILE

DATA QUALITY



## DataPlex

The screenshot displays the Google Cloud DataPlex interface. The top section shows the 'Create Dataplex data quality scan' dialog box. The 'Define scan' section includes fields for 'Display name', 'ID' (a32dd4706-fa08-48ed-bfd5-6bde9aca2731), and 'Description'. The 'Table to scan' section shows the table name 'projects/pucmvp/datasets/emissao\_raw/tables/d\_emissao\_inicial'. The 'Scope' section is set to 'Entire data'. The 'Filters (optional)' section includes 'Filter rows' and 'Sampling size' (All data).

The bottom section shows the 'Schema' view of the dataset 'd\_emissao\_inicial'. The schema table is as follows:

Field name	Type	Mode	Column Tags	Policy Tags	Business Terms	Description
SIT_INT_ID_SITE	INT64	NULLABLE	+	-	+	-
CAR_INT_CD_STATUS	INT64	NULLABLE	+	-	+	-
CAR_STR_NR_CPF	INT64	NULLABLE	+	-	+	-
CAR_STR_DS_NOMELINHA1	STRING	NULLABLE	+	-	+	-
CAR_DAT_DT_NASCIMENTO	TIMESTAMP	NULLABLE	+	-	+	-
CAR_STR_ID_CATEGORIA	STRING	NULLABLE	+	-	+	-
CAR_STR_ID_TIPO	STRING	NULLABLE	+	-	+	-
CAR_STR_DS_UF	STRING	NULLABLE	+	-	+	-
CAR_DAT_DT_EMISSAO	TIMESTAMP	NULLABLE	+	-	+	-
CAR_DAT_DT_PRIMEIRAHABILITACAO	TIMESTAMP	NULLABLE	+	-	+	-
CAR_DAT_DT_VALIDADE	TIMESTAMP	NULLABLE	+	-	+	-
CAR_DBL_NR_REGISTRO	DOUBLE	NULLABLE	+	-	+	-
CAR_STR_DS_OBSERVACAOLINHA1	STRING	NULLABLE	+	-	+	-
CAR_STR_DS_OBSERVACAOLINHA2	STRING	NULLABLE	+	-	+	-

## Metabase

Started from Emissão por UF

Search...

New question

bigquery

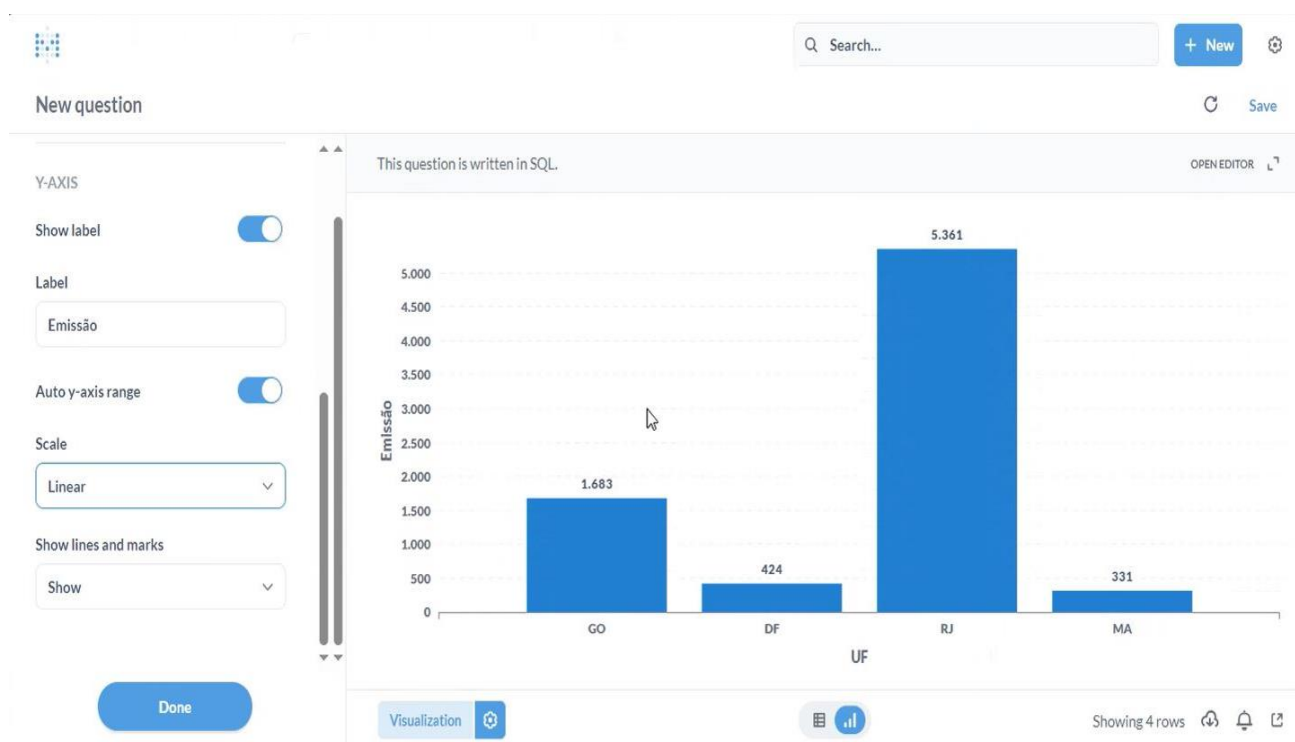
```

1 SELECT CAR_STR_DS_UF as UF, count(distinct CAR_STR_NR_CPF) qtd
2 FROM `emissao_cnh.d tst_metabase`
3 where 1=1
4 [[and {{UF}}]]
5 group by 1

```

An error occurred in your query

400 Bad Request POST  
<https://www.googleapis.com/bigquery/v2/projects/queries>  
 { "code": 400, "errors": [ { "domain": "global", "location": "q",  
 "locationType": "parameter", "message": "Unrecognized name:



## Respostas

### 1- Total Emissão por UF

```
In [12]: # Emissao de Ct's por UF
dframe_filtrado_1 = dframe_filtrado.groupby(["DS_UF"]).agg(F.count("NR_CPF").alias("QTD_CNH"))
dframe_filtrado_1.show()
```

[Stage 21:>

(0 + 1) / 1]

DS_UF	QTD_CNH
GO	1683
RJ	5361
MA	331
DF	424

