



Salário dos Brasileiros

O Poder de Compra do Servidor Público da
Região Sudeste

SoulCode Academy, 2021

Equipe



Débora Azevedo



Igor Cruz



Rodrigo Silva



Sabrina Lopes



OBJETIVO DO PROJETO

- Analisar o poder de compra do servidor público da região Sudeste
- Impacto da inflação no poder de compra, de acordo com dados sobre salário
- Poder de compra



CONJUNTO DE DADOS

- Portal de transparência das capitais
- IBGE
- IPEAData



CONTEÚDO DOS DADOS

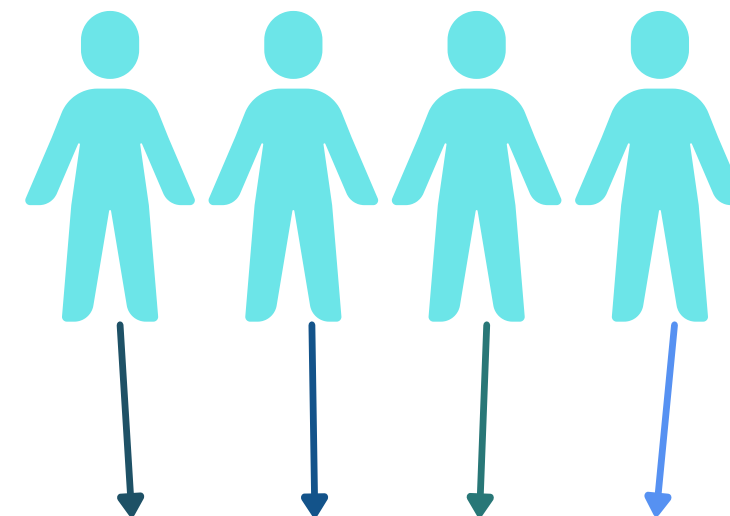
- Tabelas dos anos de 2020 e 2021
- São Paulo, Rio de Janeiro, Minas Gerais e Espírito Santo
- IPCA nacional e a nível estadual
- PIB Nacional



METODOLOGIA ÁGIL



Trabalho realizado a partir
de pequenos ciclos de
atividades



MG

SP

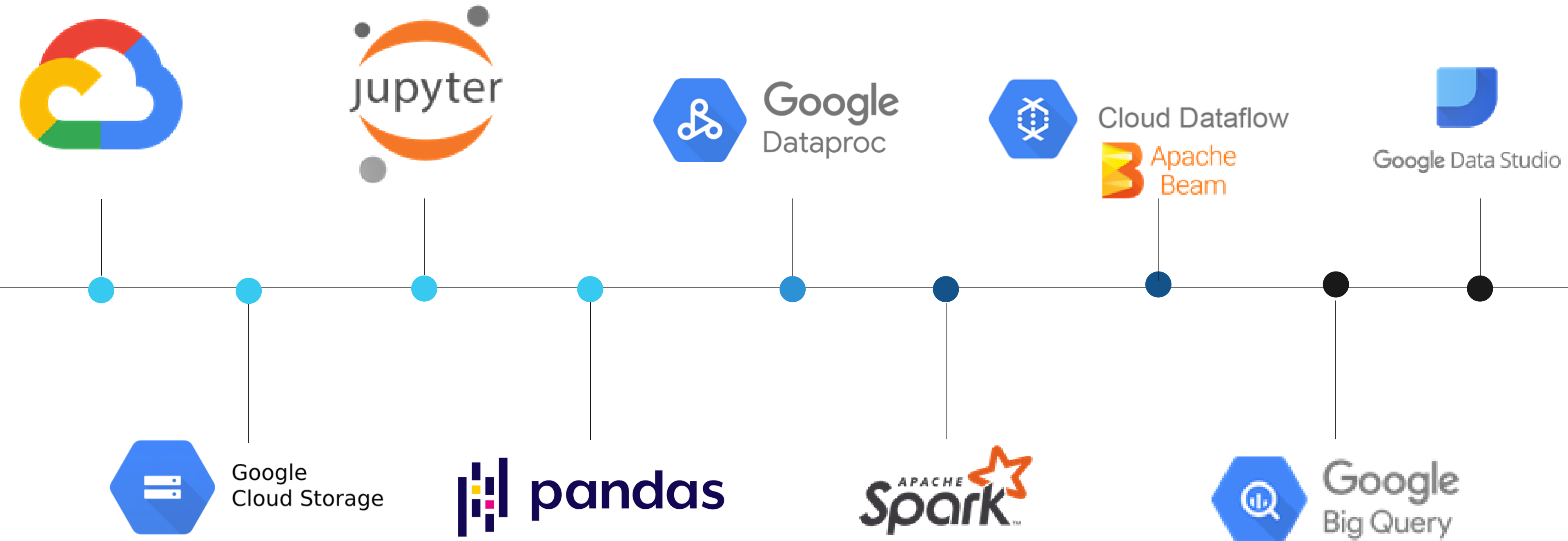
ES

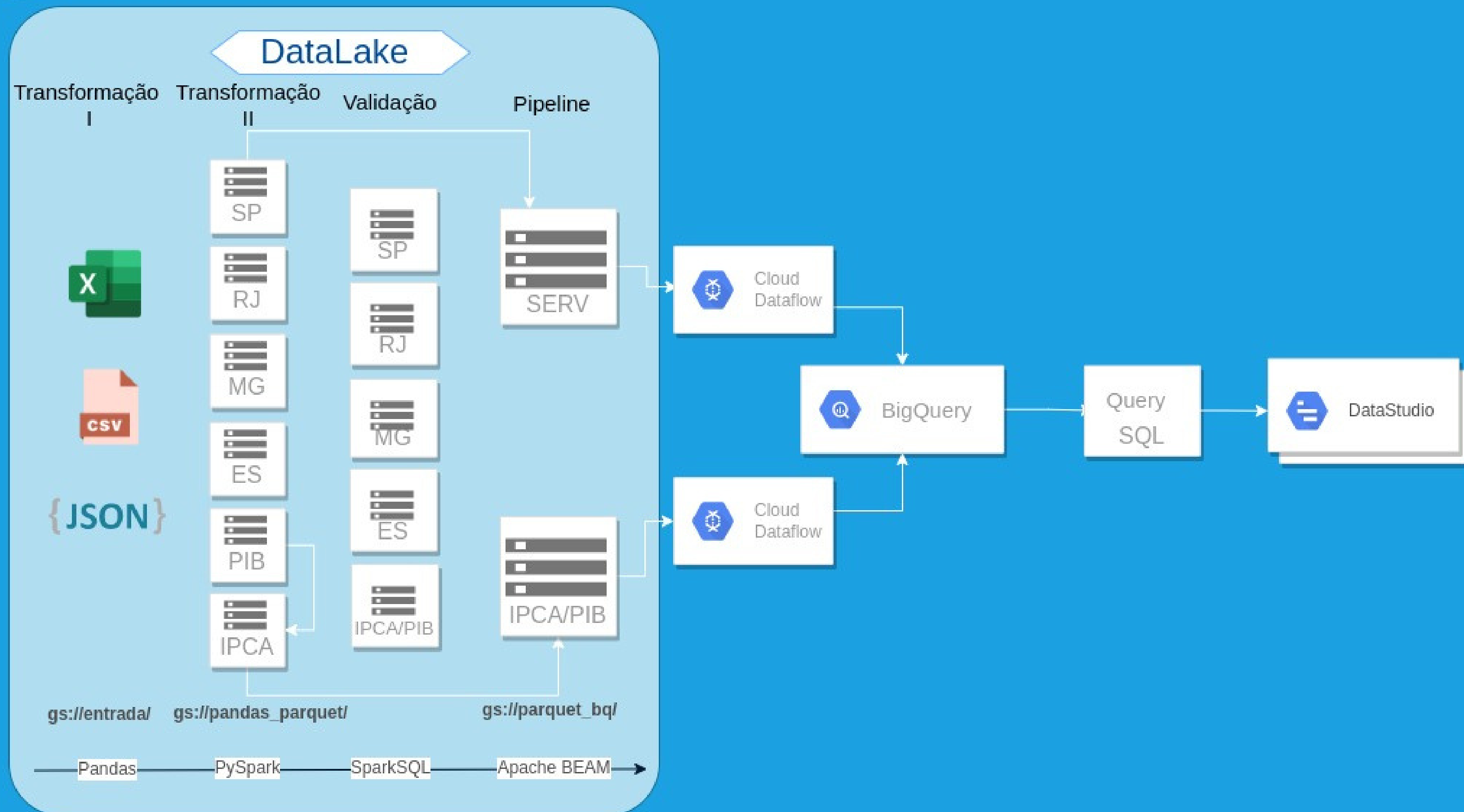
RJ



Trello

FERRAMENTAS UTILIZADAS







ETAPAS DO PROCESSO



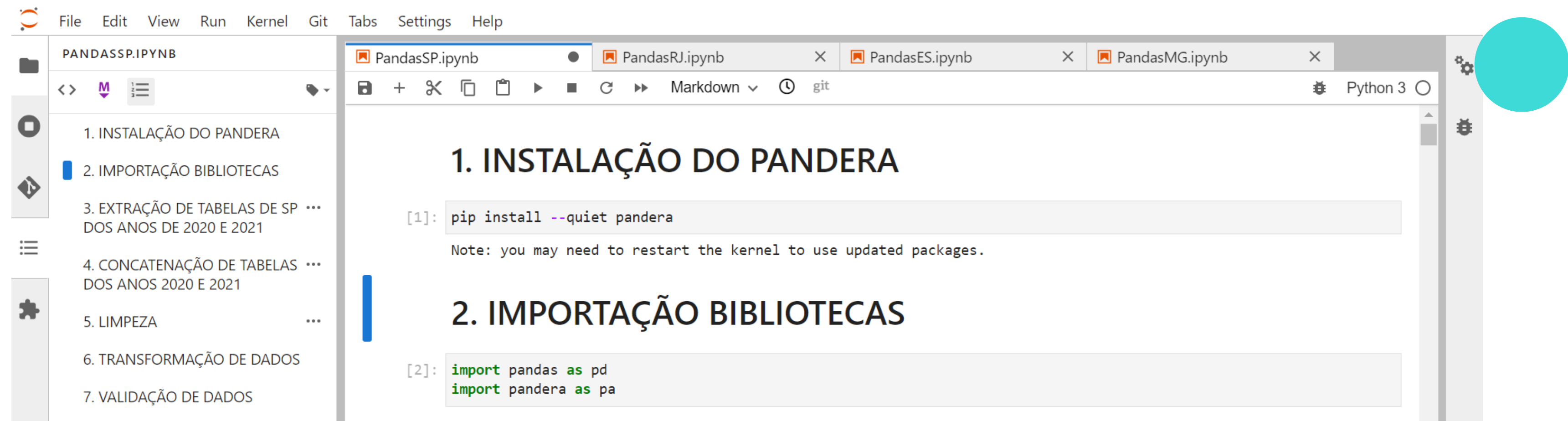
- Extração de dados
- Concatenação
- Verificação de inconsistências
- Limpeza de dados
- Transformações
- Otimização dos dados
- Exportação dos arquivos na extensão parquet



pandera

- Validação dos dados

Instalação e importação de bibliotecas



File Edit View Run Kernel Git Tabs Settings Help

PANDASSP.IPYNB

1. INSTALAÇÃO DO PANDERA

2. IMPORTAÇÃO BIBLIOTECAS

3. EXTRAÇÃO DE TABELAS DE SP ...
DOS ANOS DE 2020 E 2021

4. CONCATENAÇÃO DE TABELAS ...
DOS ANOS 2020 E 2021

5. LIMPEZA ...

6. TRANSFORMAÇÃO DE DADOS

7. VALIDAÇÃO DE DADOS

PandasSP.ipynb PandasRJ.ipynb PandasES.ipynb PandasMG.ipynb

Python 3

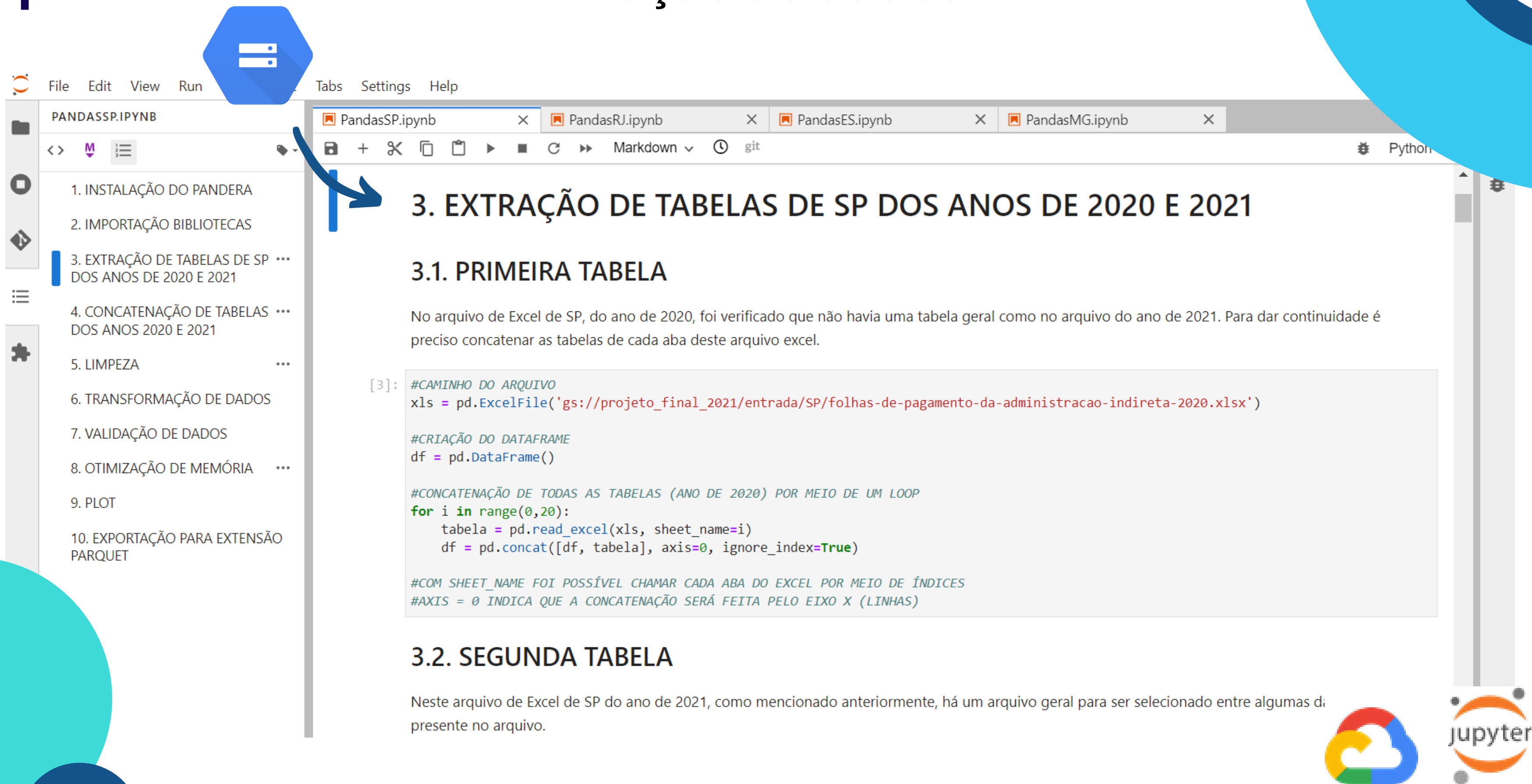
1. INSTALAÇÃO DO PANDERA

```
[1]: pip install --quiet pandera
```

Note: you may need to restart the kernel to use updated packages.

2. IMPORTAÇÃO BIBLIOTECAS

```
[2]: import pandas as pd  
import pandera as pa
```



File Edit View Run Tabs Settings Help

PANDASSP.IPYNB

- 1. INSTALAÇÃO DO PANDERA
- 2. IMPORTAÇÃO BIBLIOTECAS
- 3. EXTRAÇÃO DE TABELAS DE SP DOS ANOS DE 2020 E 2021
- 4. CONCATENAÇÃO DE TABELAS DOS ANOS 2020 E 2021
- 5. LIMPEZA
- 6. TRANSFORMAÇÃO DE DADOS
- 7. VALIDAÇÃO DE DADOS
- 8. OTIMIZAÇÃO DE MEMÓRIA
- 9. PLOT
- 10. EXPORTAÇÃO PARA EXTENSÃO PARQUET

3. EXTRAÇÃO DE TABELAS DE SP DOS ANOS DE 2020 E 2021

3.1. PRIMEIRA TABELA

No arquivo de Excel de SP, do ano de 2020, foi verificado que não havia uma tabela geral como no arquivo do ano de 2021. Para dar continuidade é preciso concatenar as tabelas de cada aba deste arquivo excel.

```
[3]: #CAMINHO DO ARQUIVO
xls = pd.ExcelFile('gs://projeto_final_2021/entrada/SP/folhas-de-pagamento-da-administracao-indireta-2020.xlsx')



#CRIAÇÃO DO DATAFRAME
df = pd.DataFrame()

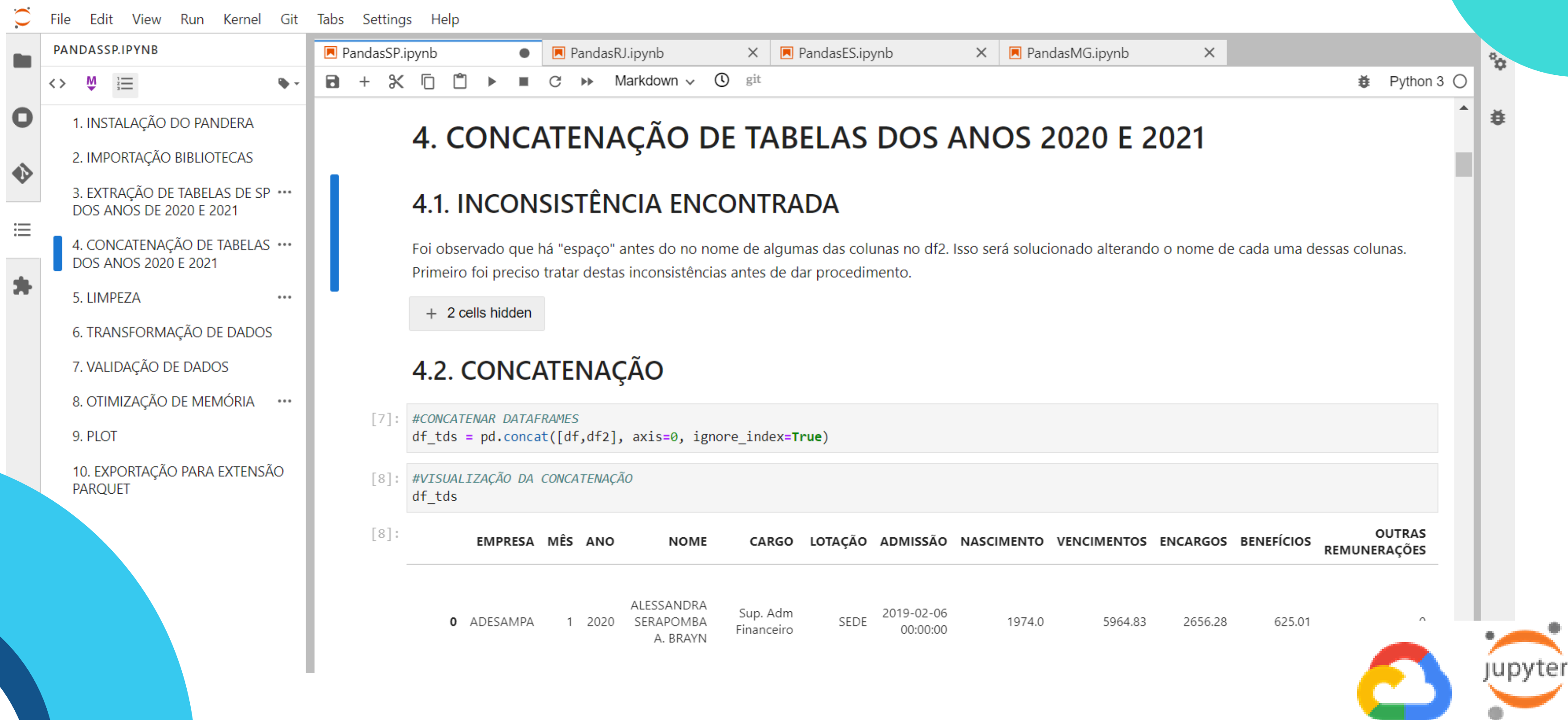
#CONCATENAÇÃO DE TODAS AS TABELAS (ANO DE 2020) POR MEIO DE UM LOOP
for i in range(0,20):
    tabela = pd.read_excel(xls, sheet_name=i)
    df = pd.concat([df, tabela], axis=0, ignore_index=True)

#COM SHEET_NAME FOI POSSÍVEL CHAMAR CADA ABA DO EXCEL POR MEIO DE ÍNDICES
#AXIS = 0 INDICA QUE A CONCATENAÇÃO SERÁ FEITA PELO EIXO X (LINHAS)
```

3.2. SEGUNDA TABELA

Neste arquivo de Excel de SP do ano de 2021, como mencionado anteriormente, há um arquivo geral para ser selecionado entre algumas di: presente no arquivo.





The image shows a JupyterLab interface with a notebook titled "PANDASSP.IPYNB". The notebook has four tabs: "PandasSP.ipynb", "PandasRJ.ipynb", "PandasES.ipynb", and "PandasMG.ipynb". The active tab is "PandasSP.ipynb". The notebook content is as follows:

4. CONCATENAÇÃO DE TABELAS DOS ANOS 2020 E 2021

4.1. INCONSISTÊNCIA ENCONTRADA

Foi observado que há "espaço" antes do no nome de algumas das colunas no df2. Isso será solucionado alterando o nome de cada uma dessas colunas. Primeiro foi preciso tratar destas inconsistências antes de dar procedimento.

+ 2 cells hidden

4.2. CONCATENAÇÃO

```
[7]: #CONCATENAR DATAFRAMES
df_tds = pd.concat([df,df2], axis=0, ignore_index=True)
```

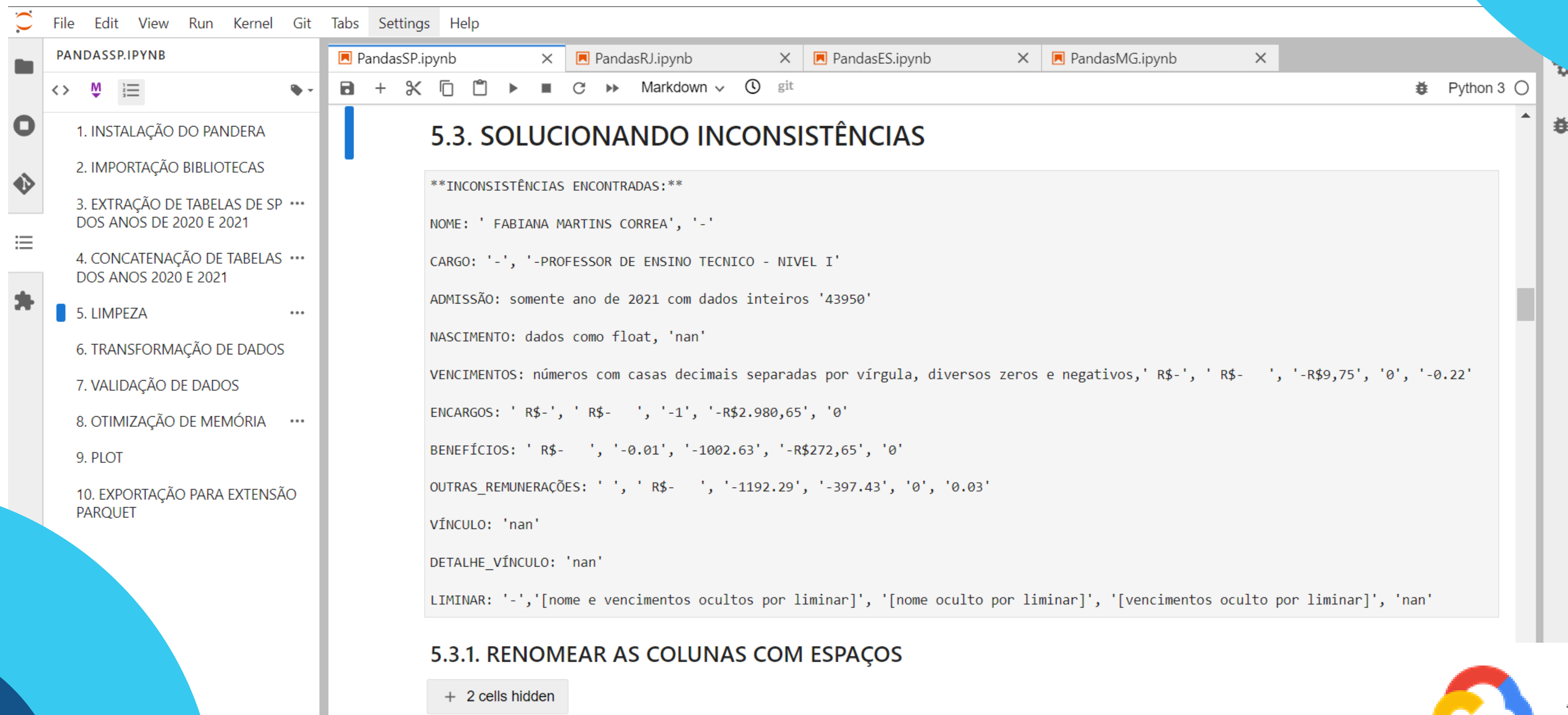
```
[8]: #VISUALIZAÇÃO DA CONCATENAÇÃO
df_tds
```

```
[8]:
```

	EMPRESA	MÊS	ANO	NOME	CARGO	LOTAÇÃO	ADMISSÃO	NASCIMENTO	VENCIMENTOS	ENCARGOS	BENEFÍCIOS	OUTRAS REMUNERAÇÕES
0	ADESAMPA	1	2020	ALESSANDRA SERAPOMBA A. BRAYN	Sup. Adm Financeiro	SEDE	2019-02-06 00:00:00	1974.0	5964.83	2656.28	625.01	^

Logos for Google Cloud and Jupyter are visible in the bottom right corner.

Verificação de inconsistências e Limpeza de dados



The image shows a JupyterLab interface with a sidebar on the left containing a table of contents. The main area displays a notebook titled '5.3. SOLUCIONANDO INCONSISTÊNCIAS'. The notebook content lists various data inconsistencies found in a dataset, such as missing values, incorrect data types, and formatting issues. Below the list, a sub-section '5.3.1. RENOMEAR AS COLUNAS COM ESPAÇOS' is visible, followed by a button indicating that 2 cells are hidden.

5.3. SOLUCIONANDO INCONSISTÊNCIAS

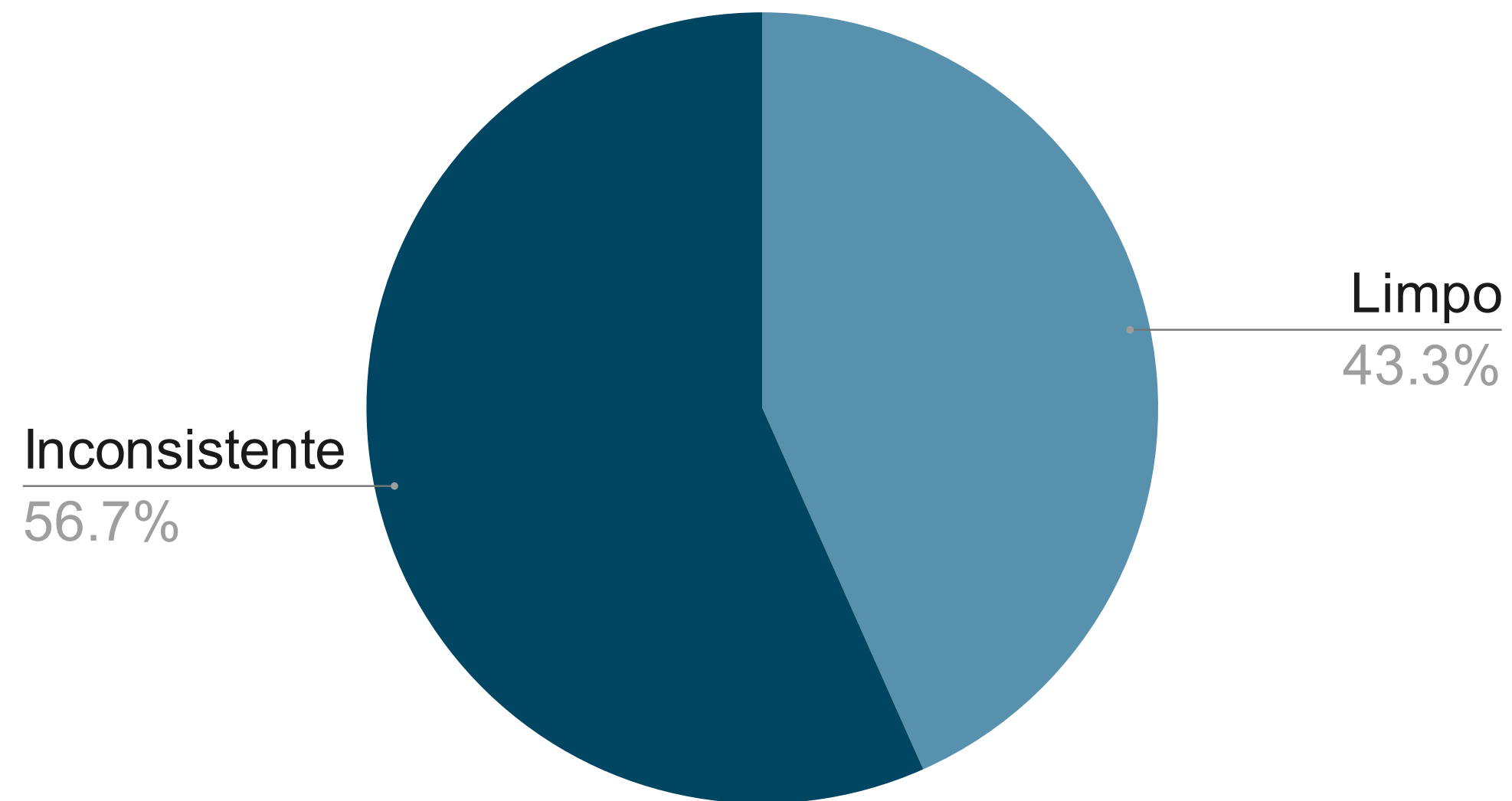
****INCONSISTÊNCIAS ENCONTRADAS:****

- NOME: ' FABIANA MARTINS CORREA', '-'
- CARGO: '-', '-PROFESSOR DE ENSINO TECNICO - NIVEL I'
- ADMISSÃO: somente ano de 2021 com dados inteiros '43950'
- NASCIMENTO: dados como float, 'nan'
- VENCIMENTOS: números com casas decimais separadas por vírgula, diversos zeros e negativos, ' R\$-', ' R\$- ', '-R\$9,75', '0', '-0.22'
- ENCARGOS: ' R\$-', ' R\$- ', '-1', '-R\$2.980,65', '0'
- BENEFÍCIOS: ' R\$- ', '-0.01', '-1002.63', '-R\$272,65', '0'
- OUTRAS_REMUNERAÇÕES: ' ', ' R\$- ', '-1192.29', '-397.43', '0', '0.03'
- VÍNCULO: 'nan'
- DETALHE_VÍNCULO: 'nan'
- LIMINAR: '-', '[nome e vencimentos ocultos por liminar]', '[nome oculto por liminar]', '[vencimentos oculto por liminar]', 'nan'

5.3.1. RENOMEAR AS COLUNAS COM ESPAÇOS

+ 2 cells hidden

QUALIDADE DOS DADOS COLETADOS





The image shows a JupyterLab interface with a sidebar on the left containing a table of contents. The main area displays a notebook titled 'PandasSP.ipynb' with the following content:

6. TRANSFORMAÇÃO DE DADOS

Como nos passos anteriores foi preciso colocar todas as colunas para string afim de solucionar as inconsistências, nesta etapa, antes da validação dos dados utilizando o Pandera, foi necessário transformar o tipo de dado de colunas específicas, como indicado abaixo.

```
[26]: #TRANSFORMAR COLUNAS PARA INT
      colunas_int = ["MÊS", "ANO", "NASCIMENTO"]

      for C in colunas_int:
          df_tds[C] = pd.to_numeric(df_tds[C])

      #COLUNAS PARA FLOAT
      colunas_float = ["VENCIMENTOS", "ENCARGOS", "BENEFÍCIOS", "OUTRAS_REMUNERAÇÕES"]

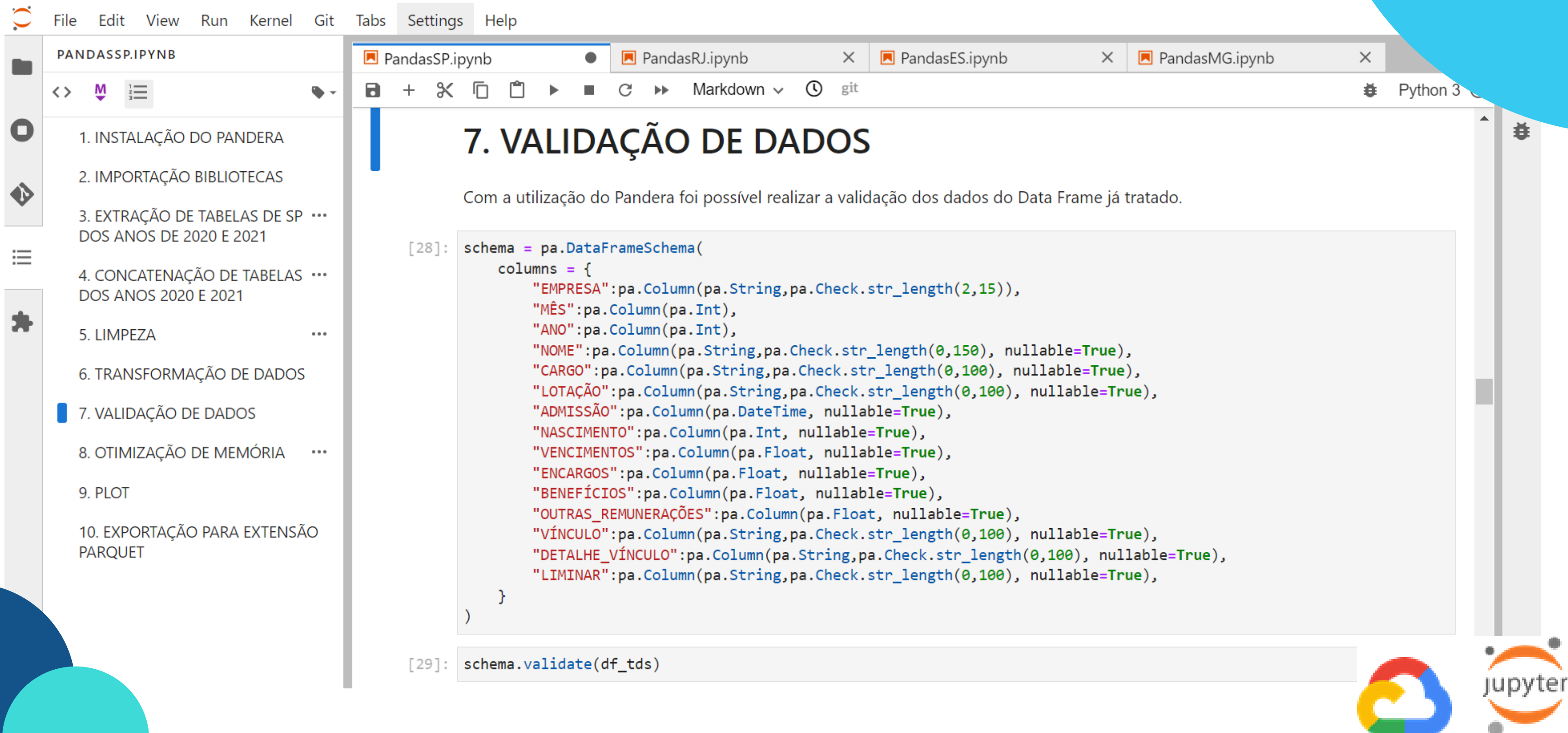
      for C in colunas_float:
          df_tds[C] = pd.to_numeric(df_tds[C])

      #COLUNA PARA DATETIME
      df_tds["ADMISSÃO"] = pd.to_datetime(df_tds["ADMISSÃO"])

[27]: df_tds.dtypes
      ...
```

The sidebar on the left lists the following steps:

1. INSTALAÇÃO DO PANDERA
2. IMPORTAÇÃO BIBLIOTECAS
3. EXTRAÇÃO DE TABELAS DE SP ... DOS ANOS DE 2020 E 2021
4. CONCATENAÇÃO DE TABELAS ... DOS ANOS 2020 E 2021
5. LIMPEZA ...
6. TRANSFORMAÇÃO DE DADOS
7. VALIDAÇÃO DE DADOS
8. OTIMIZAÇÃO DE MEMÓRIA ...
9. PLOT
10. EXPORTAÇÃO PARA EXTENSÃO PARQUET



The image shows a JupyterLab interface with a notebook titled "7. VALIDAÇÃO DE DADOS". The left sidebar contains a table of contents with 10 items, where item 7 is highlighted. The notebook has four tabs: "PandasSP.ipynb", "PandasRJ.ipynb", "PandasES.ipynb", and "PandasMG.ipynb". The active tab is "PandasSP.ipynb". The notebook content includes a title "7. VALIDAÇÃO DE DADOS", a paragraph explaining the use of Pandera for data validation, and two code cells. The first code cell defines a Pandera schema with columns for company, month, year, name, cargo, location, admission, birth date, salary, responsibilities, other remunerations, link, link details, and a limit. The second code cell calls the schema's validate method on a DataFrame.

File Edit View Run Kernel Git Tabs Settings Help

PANDASSP.IPYNB

- 1. INSTALAÇÃO DO PANDERA
- 2. IMPORTAÇÃO BIBLIOTECAS
- 3. EXTRAÇÃO DE TABELAS DE SP ... DOS ANOS DE 2020 E 2021
- 4. CONCATENAÇÃO DE TABELAS ... DOS ANOS 2020 E 2021
- 5. LIMPEZA ...
- 6. TRANSFORMAÇÃO DE DADOS
- 7. VALIDAÇÃO DE DADOS**
- 8. OTIMIZAÇÃO DE MEMÓRIA ...
- 9. PLOT
- 10. EXPORTAÇÃO PARA EXTENSÃO PARQUET

PandasSP.ipynb PandasRJ.ipynb PandasES.ipynb PandasMG.ipynb

7. VALIDAÇÃO DE DADOS

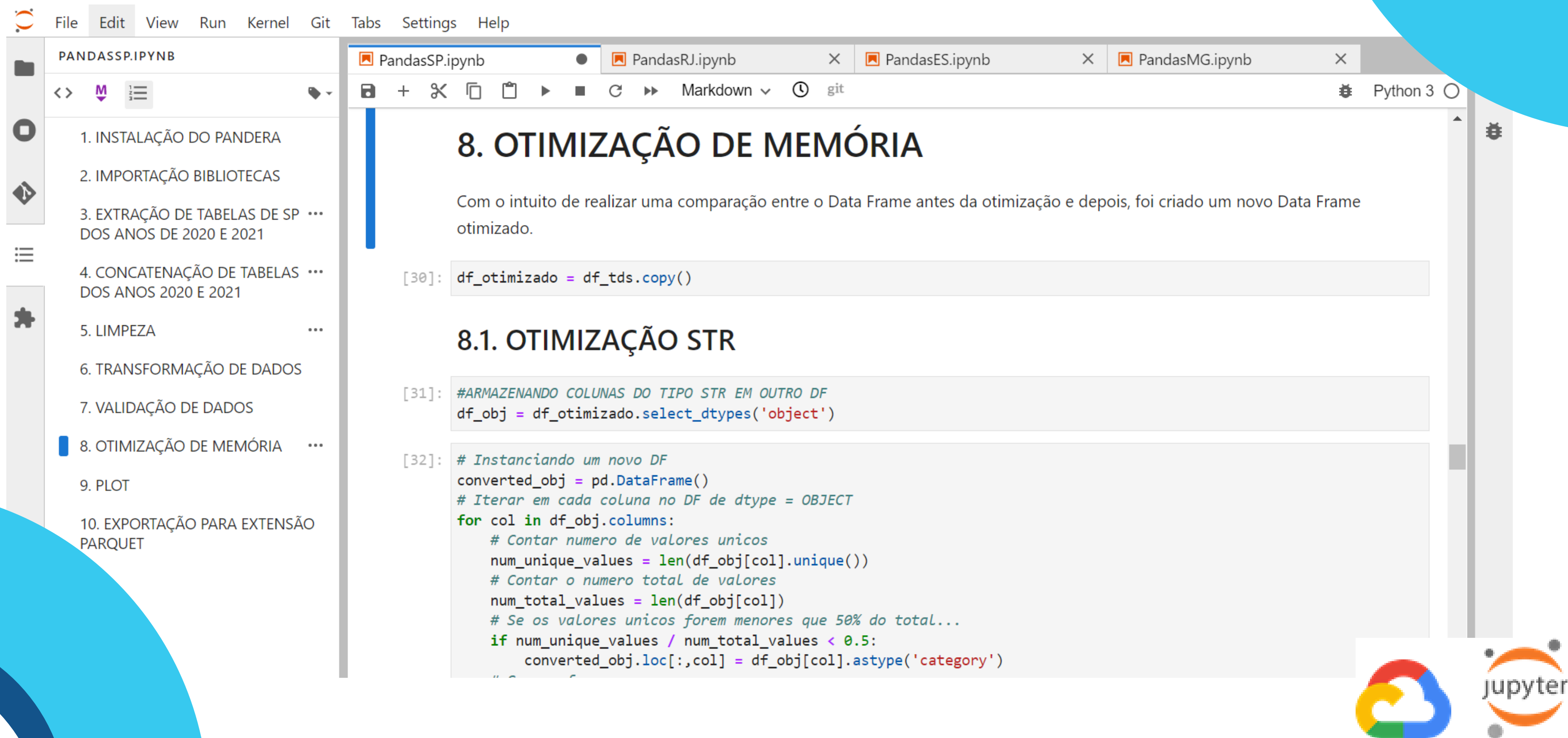
Com a utilização do Pandera foi possível realizar a validação dos dados do Data Frame já tratado.

```
[28]: schema = pa.DataFrameSchema(
    columns = {
        "EMPRESA": pa.Column(pa.String, pa.Check.str_length(2, 15)),
        "MÊS": pa.Column(pa.Int),
        "ANO": pa.Column(pa.Int),
        "NOME": pa.Column(pa.String, pa.Check.str_length(0, 150), nullable=True),
        "CARGO": pa.Column(pa.String, pa.Check.str_length(0, 100), nullable=True),
        "LOTAÇÃO": pa.Column(pa.String, pa.Check.str_length(0, 100), nullable=True),
        "ADMISSÃO": pa.Column(pa.DateTime, nullable=True),
        "NASCIMENTO": pa.Column(pa.Int, nullable=True),
        "VENCIMENTOS": pa.Column(pa.Float, nullable=True),
        "ENCARGOS": pa.Column(pa.Float, nullable=True),
        "BENEFÍCIOS": pa.Column(pa.Float, nullable=True),
        "OUTRAS_REMUNERAÇÕES": pa.Column(pa.Float, nullable=True),
        "VÍNCULO": pa.Column(pa.String, pa.Check.str_length(0, 100), nullable=True),
        "DETALHE_VÍNCULO": pa.Column(pa.String, pa.Check.str_length(0, 100), nullable=True),
        "LIMINAR": pa.Column(pa.String, pa.Check.str_length(0, 100), nullable=True),
    }
)
```

```
[29]: schema.validate(df_tds)
```

Python 3

Google Cloud Jupyter



The image shows a JupyterLab interface with a sidebar on the left containing a table of contents. The main area displays a notebook titled 'PandasSP.ipynb' with the following content:

8. OTIMIZAÇÃO DE MEMÓRIA

Com o intuito de realizar uma comparação entre o Data Frame antes da otimização e depois, foi criado um novo Data Frame otimizado.

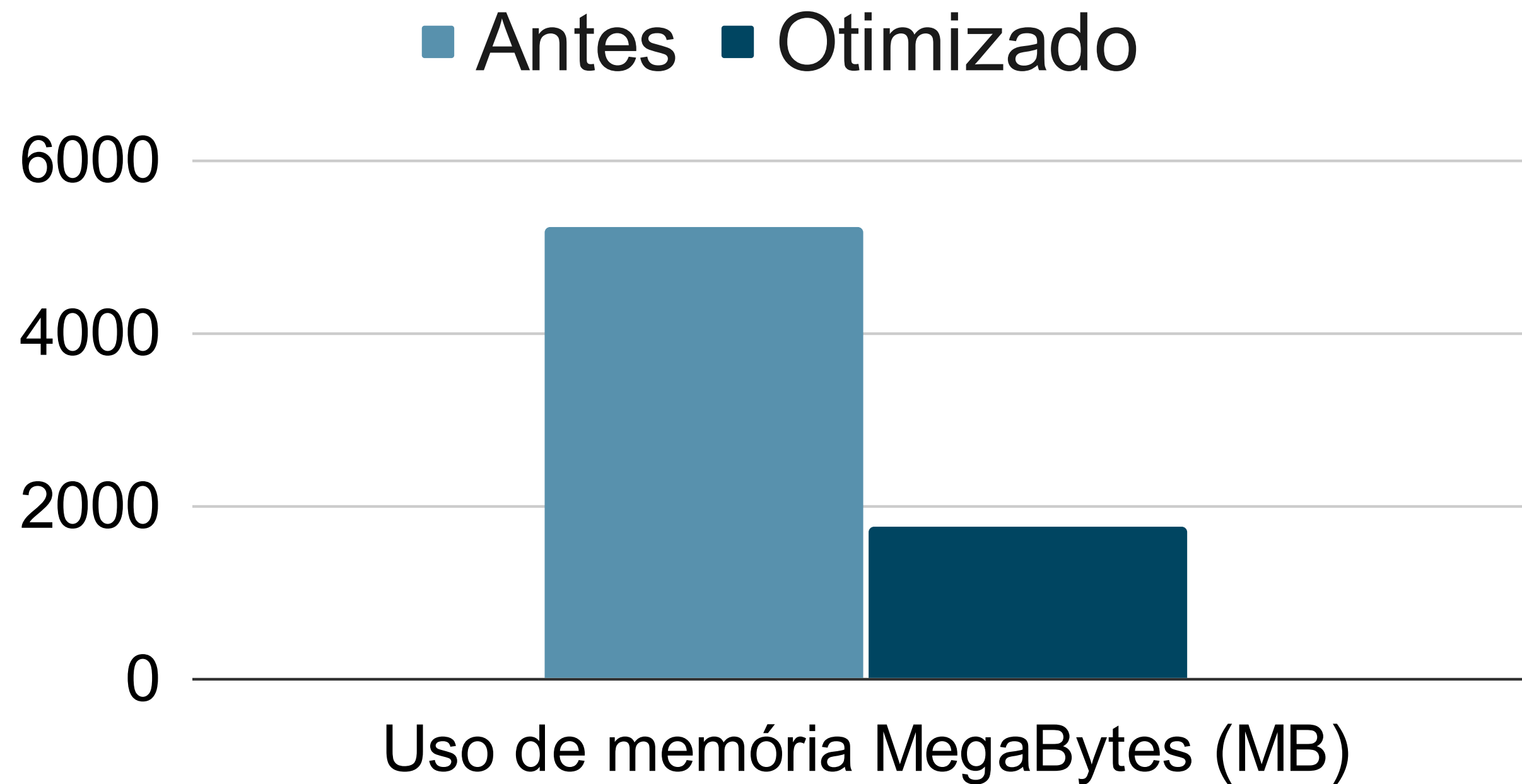
```
[30]: df_otimizado = df_tds.copy()
```

8.1. OTIMIZAÇÃO STR

```
[31]: #ARMAZENANDO COLUNAS DO TIPO STR EM OUTRO DF
df_obj = df_otimizado.select_dtypes('object')

[32]: # Instanciando um novo DF
converted_obj = pd.DataFrame()
# Iterar em cada coluna no DF de dtype = OBJECT
for col in df_obj.columns:
    # Contar numero de valores unicos
    num_unique_values = len(df_obj[col].unique())
    # Contar o numero total de valores
    num_total_values = len(df_obj[col])
    # Se os valores unicos forem menores que 50% do total...
    if num_unique_values / num_total_values < 0.5:
        converted_obj.loc[:,col] = df_obj[col].astype('category')
```

At the bottom right, there are logos for Google Cloud and Jupyter.



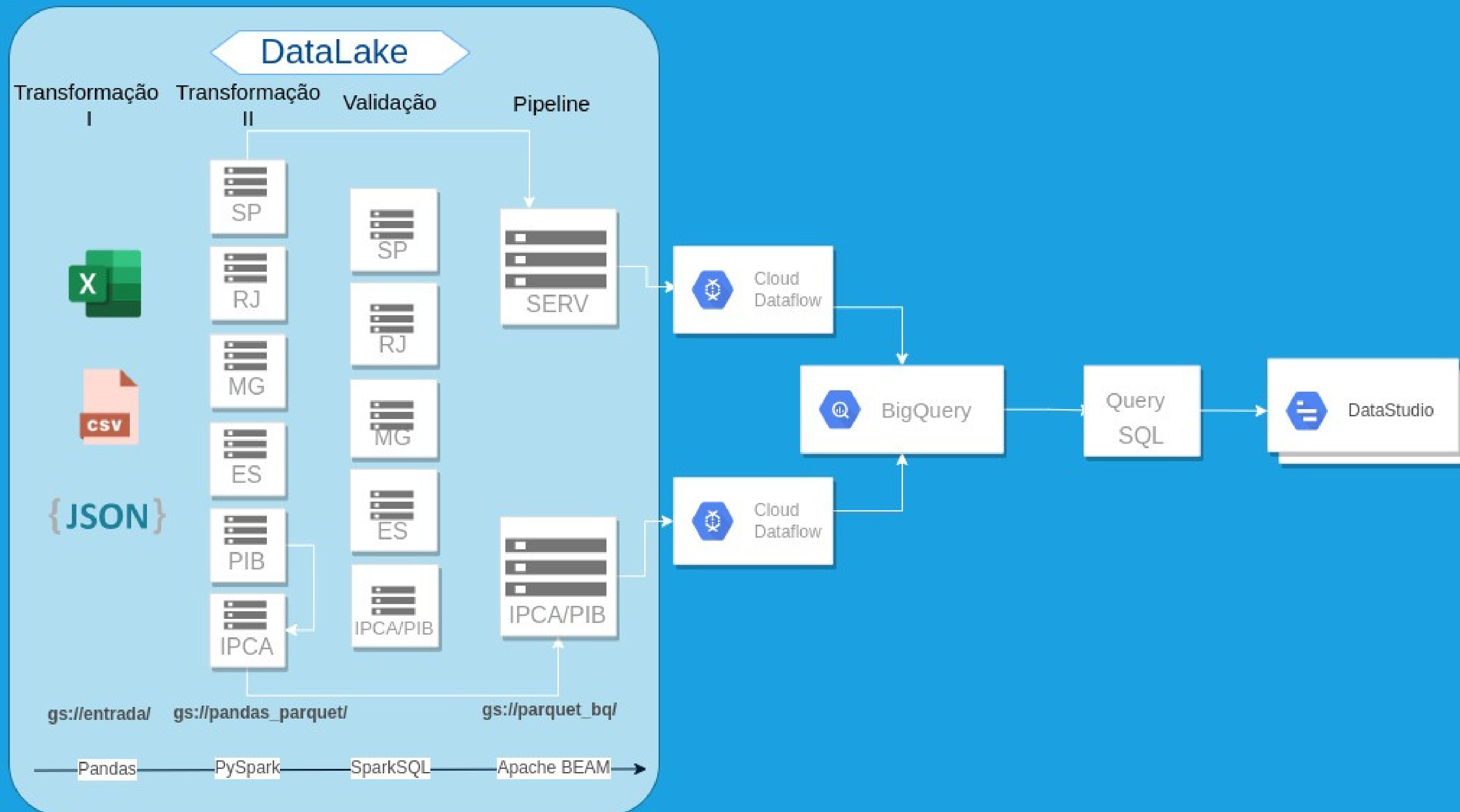
Otimização
de memória



SALVAMENTO DOS ARQUIVOS

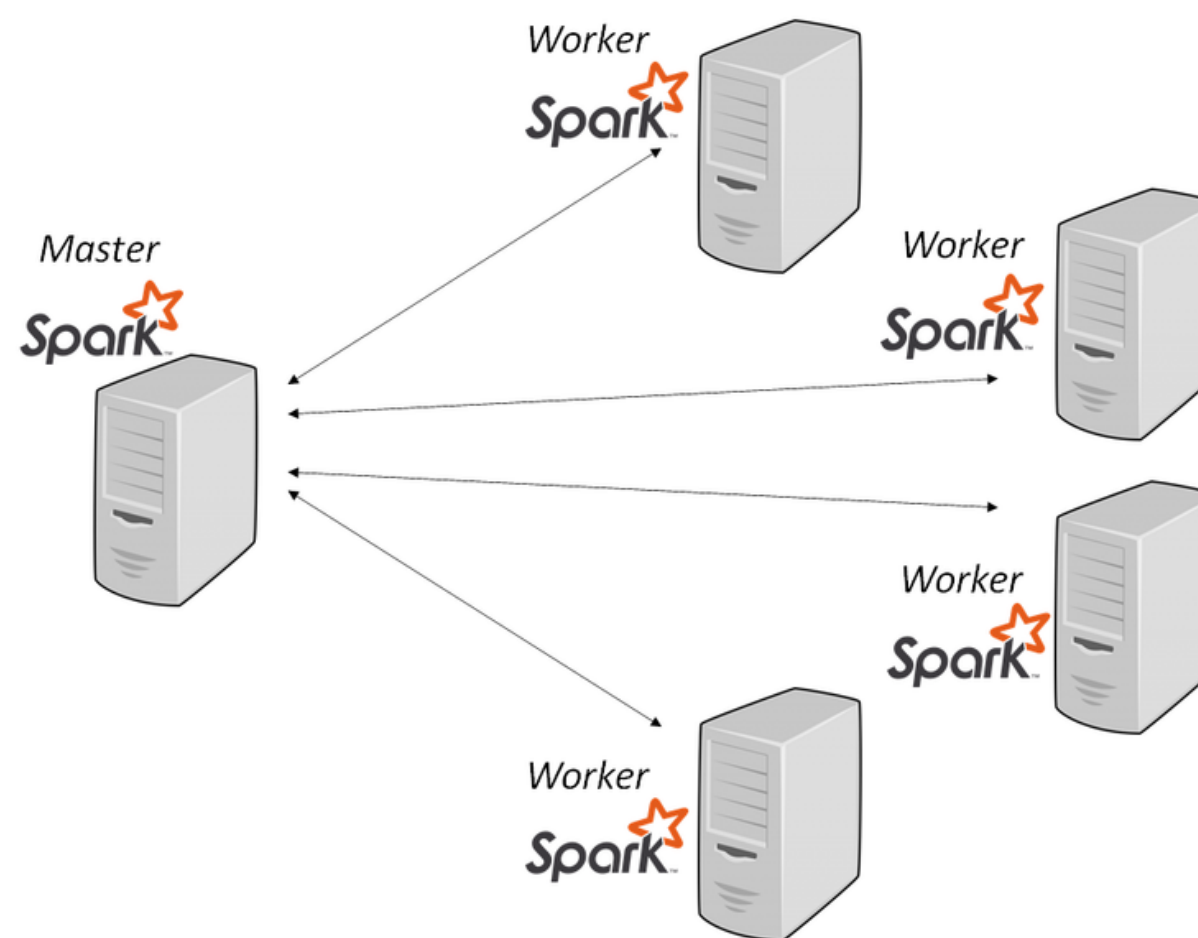


- Otimização no tempo de processamento
- Armazenamento em formato colunar
- Arquivo compactado (snappy)





Framework opensource para
processamento de dados de
forma paralela e distribuida.





- Seleção das colunas
- Criação de colunas
- Renomeação de colunas
- Ordenação colunas
- Concatenação

salario-brasileiro > pyspark-cluster Sign out

File Edit View Run Kernel Git Tabs Settings Help

PYSPARK_SP.IPYNB

- 1. IMPORTAÇÃO DAS BIBLIOTECAS
- 2. CRIAÇÃO DA SPARK SESSION
- 3. EXTRAÇÃO DE DADOS DE SP
- 4. ORDENANDO, CRIANDO E ALTERANDO COLUNAS
 - 4.1. CRIAÇÃO DA COLUNA SALARIOLÍQUIDO
 - 4.2. FILTRANDO COLUNAS NECESSÁRIAS PARA ANÁLISE
 - 4.3. PADRONIZANDO NOME DAS COLUNAS
 - 4.4. PADRONIZANDO DADOS DA COLUNA SITUACAO
 - 4.5. ORDENANDO COLUNAS
- 5. VISUALIZAÇÃO TABELA FINAL DE SP
- 6. EXPORTAÇÃO DO ARQUIVO

1. IMPORTAÇÃO DAS BIBLIOTECAS

```
[1]: from pyspark.sql import SparkSession
from pyspark.sql.functions import *
from pyspark.sql.types import *
from pyspark.sql.window import Window as W
```

2. CRIAÇÃO DA SPARK SESSION

```
[ ]: spark = (SparkSession.builder
    .master("local")
    .appName("Pyspark_SP")
    .config("spark.ui.port", "4050")
    .getOrCreate())
```

[3]: spark

[3]: SparkSession - hive

SparkContext

Spark UI

Version v3.1.2
Master local
AppName Pyspark_SP

3. EXTRAÇÃO DE DADOS DE SP

Simple 0 4 PySpark | Idle Saving completed Mode: Command Ln 1, Col 1 Pyspark_SP.ipynb



Spark SQL



- Consultas SQL
- Validação do dataframe
- Inconsistência nos dados!

salario-brasileiro > pyspark-cluster Sign out

File Edit View Run Kernel Git Tabs Settings Help

SQL_SP.IPYNB

1. Importação das bibliotecas

2. Consultas para validação

2. Consultas para validação

[5]: *#CONSULTA DA DESCRIÇÃO DA TABELA*
`spark.sql('DESCRIBE sp').show()`

ivysettings.xml file not found in HIVE_HOME or HIVE_CONF_DIR,/etc/hive/conf.dist/ivysettings.xml will be used

col_name	data_type	comment
mes	tinyint	null
ano	smallint	null
nome	string	null
cargo	string	null
situacao	string	null
salarioLiquido	float	null
estado	string	null

[18]: *#Quantidade de servidores por cargo*
`spark.sql('SELECT cargo, COUNT(nome) AS Qtd_servidores FROM sp GROUP BY cargo').show()`

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

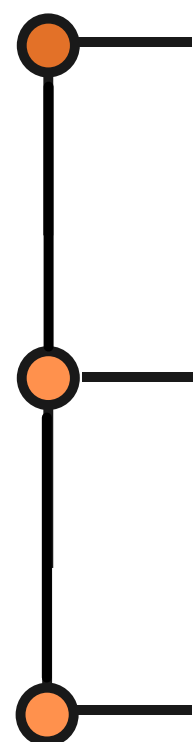
cargo	Qtd_servidores
Diretor de Divisã...	182
TEC PROJ OBRAS VI	43
ANAL DE COMUNICACAO	24
DIR PART E REPR EMPR	21
Cantor de Coral	3
SÇ TEC DE GASTROE...	1
GER ANALISE TECNICA	8
Assistente de Saú...	104
AST ADM I	171
TECNICO EM SAUDE	202

Simple 0 8 Python 3 | Idle Saving completed Mode: Command Ln 1, Col 29 SQL_SP.ipynb





Cloud Dataflow



Modelo de Pipeline
para tabela
IPCA_PIB

Modelo de Pipeline
para tabela
servidores

Extração pra o
BigQuery

```
File Edit View Run Kernel Git Interactive Beam Tabs Settings Help
IPCA_PIPELINE.IPYNB
1. Importação das bibliotecas
2. Pipeline Tabela IPCA_PIB
  2.1. Schema tabela
  2.2. Extração para o BigQuery
3. Pipeline Tabela servidores

IPCA_pipeline.ipynb
1. Apache Beam 2.33.0 for Python 3

2. Pipeline Tabela IPCA_PIB

2.1. Schema tabela

[2]: table_spec = bigquery.TableReference(
      projectId='salario-brasileiro',
      datasetId='br_me_rais',
      tableId='ipca_pib_nacional')

      SCHEMA = '''mes:INTEGER,
                  ano:INTEGER,
                  estado:STRING,
                  item:STRING,
                  IPCA_item:FLOAT,
                  IPCA_VarMensal:FLOAT,
                  IPCA_VarAcu_3Meses:FLOAT,
                  IPCA_VarAcu_6Meses:FLOAT,
                  IPCA_VarAcu_12Meses:FLOAT,
                  IPCA_VarAcu_Ano:FLOAT,
                  PIB_nacional:FLOAT'''

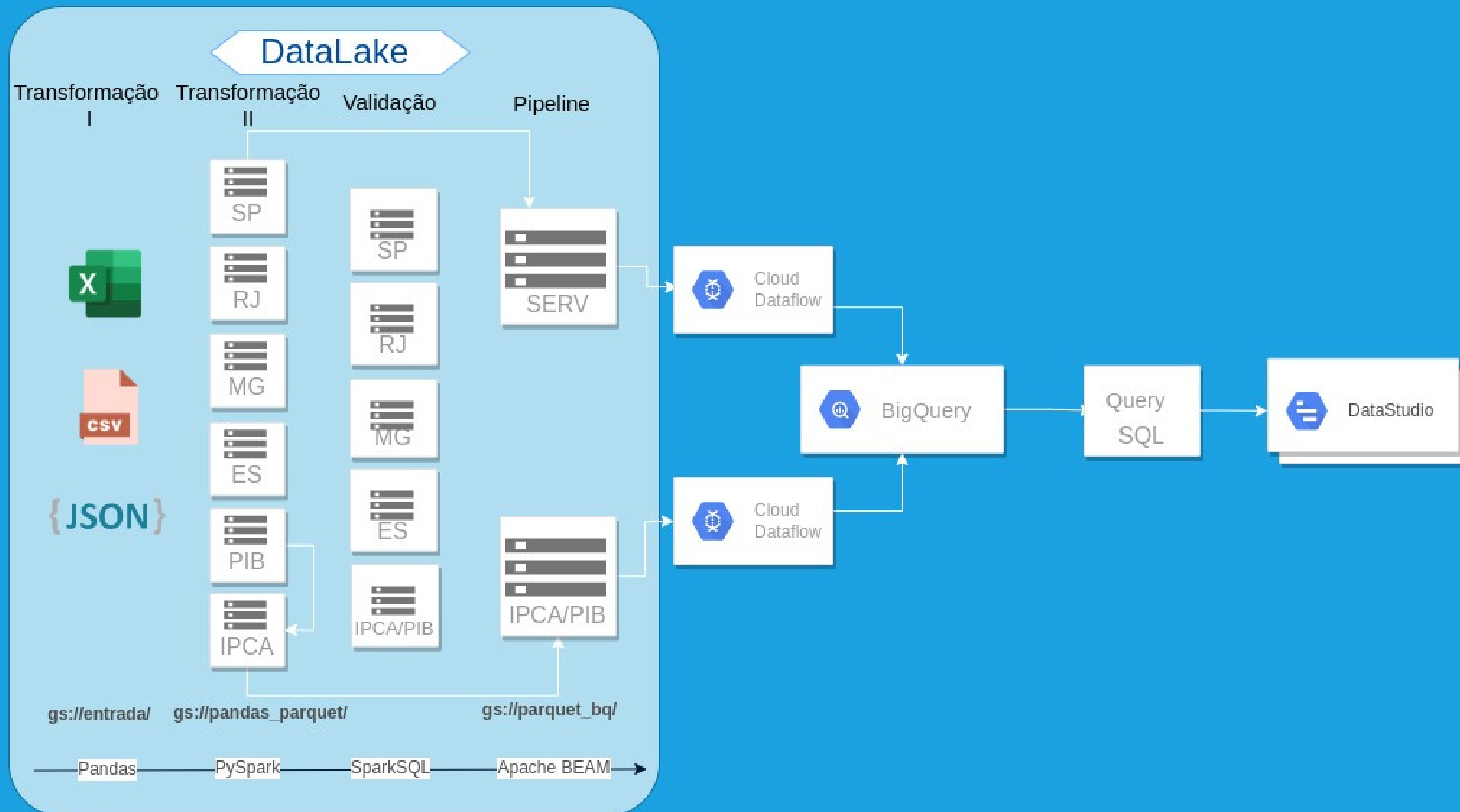
2.2. Extração para o BigQuery

[ ]: p = beam.Pipeline()

      (p | 'ReadData' >> beam.io.ReadFromParquet('gs://projeto_final_2021/parquet_to_bq/ipca_pib_nacional/ipca_pib_nacional.pa
      | 'WriteToBigQuery' >> beam.io.WriteToBigQuery(table_spec,
                                                    custom_gcs_temp_location='gs://projeto_final_2021/temp_location/',
                                                    schema=SCHEMA,
                                                    method='FILE_LOADS',
                                                    write_disposition=beam.io.BigQueryDisposition.WRITE_TRUNCATE,
                                                    create_disposition=beam.io.BigQueryDisposition.CREATE_IF_NEEDED)
      )

      p.run()
```







Google
BigQuery

Como aplicar tudo isso e conectar a tecnologia ao mundo dos negócios ?

Consolidamos todas as estruturas de dados da origem, num grande universo de dados



A partir dele podemos montar consultas simples, ou complexas, online ou em lote , ou até inserir dados de forma massiva

O cenário de negócio :

Impacto da inflação no salário do servidor público

Dados de IPCA

Dados de PIB

Salário dos Servidores



Google
BigQuery

Geramos uma estrutura única de dados a partir da qual produzimos novos insights

1

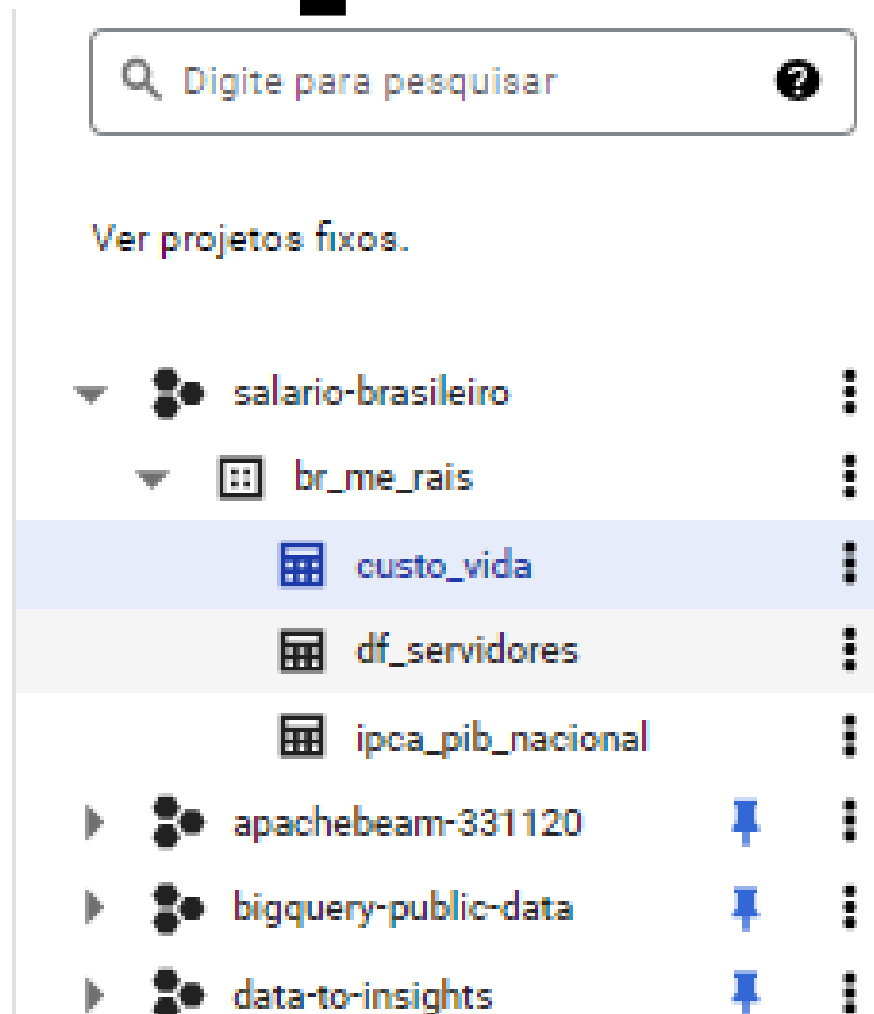
Simples

```
1  --Média salarial por cargo
2
3  SELECT
4      cargo,
5      ROUND(AVG(salarioLiquido), 2) as media_salarial
6  FROM
7      `salario-brasileiro.br_me_rais.df_servidores`
8  GROUP BY
9      cargo
10
```


Médio

```
1  -- Setor com maior índice de inflação de SP de agosto de 2021
2  SELECT
3      item,
4      IPCA_item
5  FROM
6      `salario-brasileiro.br_me_rais.ipca_pib_nacional`
7  WHERE
8      ano = 2021
9      AND mes = 8
10     AND estado = "SP"
11     AND IPCA_item IN (
12         SELECT
13             MAX(IPCA_item)
14         FROM
15             `salario-brasileiro.br_me_rais.ipca_pib_nacional`
16         WHERE
17             ano = 2021
18             AND mes = 8
19             AND estado = "SP"
20     )
```

Complexa



```
1  -- ANÁLISE DO CUSTO DE VIDA DO SERVIDOR PÚBLICO DA REGIÃO SUDESTE DO BRASIL
2  -- Nesta query queremos calcular a média dos salário dos servidores públicos
3  -- comparados ao IPCA (inflação) e PIB (produtividade).
4  CREATE OR REPLACE TABLE
5  `br_me_rais.custo_vida` AS (
6  SELECT
7    T1.*,
8    RANK() OVER(ORDER BY T1.media_salario_servidor DESC) AS ranking_salarios
9  FROM (
10   SELECT
11     T1.ano,
12     T1.mes,
13     T1.estado,
14     T2.item,
15     AVG(T2.IPCA_item) AS mediaMes_IPCA_item,
16     AVG(T2.IPCA_VarMensal) AS mediaMes_IPCA_nacional,
17     AVG(T2.PIB_nacional) AS media_PIB,
18     ROUND(AVG(T1.salarioLiquido), 2) AS media_salario_servidor
19   FROM
20     `salario-brasileiro.br_me_rais.df_servidores` AS T1
21   LEFT JOIN
22     `salario-brasileiro.br_me_rais.ipca_pib_nacional` AS T2
23   ON
24     (T1.ano,
25      T1.mes,
26      T1.estado) = (T2.ano,
27                    T2.mes,
28                    T2.estado)
29   WHERE
30     T1.situacao = 'ATIVO'
31     AND T1.situacao = 'OUTRO'
32     AND T1.salarioLiquido IS NOT NULL
33   GROUP BY
34     T1.ano,
35     T1.mes,
36     T1.estado,
37     T2.item ) AS T1
38   ORDER BY
39     media_salario_servidor DESC)
```

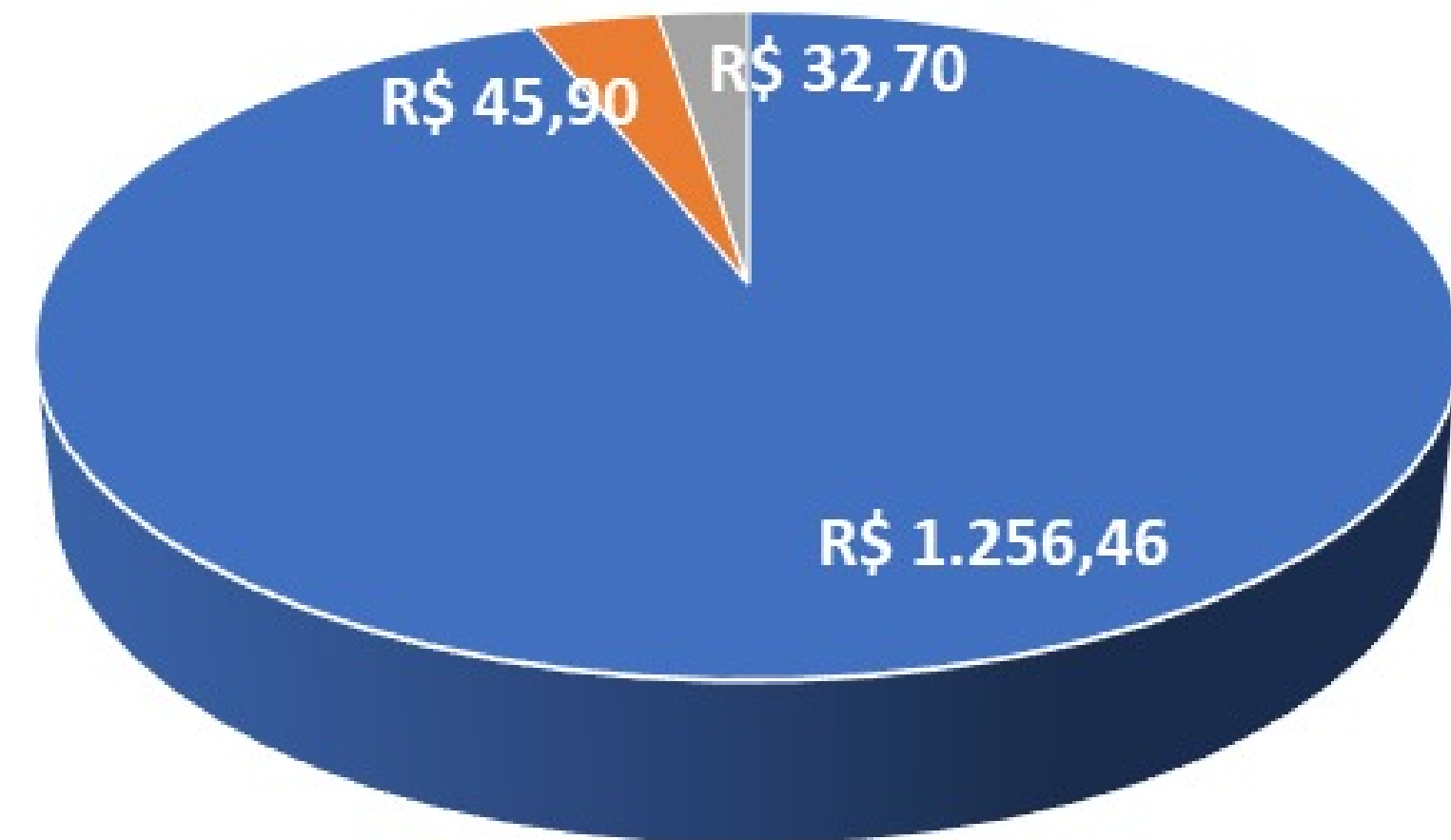


Dimensões

Time	4	pessoas
Prazo	2	semanas
Horas	96	por pessoa
Total	384	horas consumidas

Trabalhos efetuados entre 17/11 a 30/11/2021

Gastos com infraestrutura



■ Compute Engine ■ Cloud Storage ■ Notebooks

Total R\$ 1335,06

REFERÊNCIAS

Densmore, J. Data Pipelines Pocket Reference. Moving and Processing Data for Analytics. O'Reilly 2021

Wickham, H. . (2014). Tidy Data. Journal of Statistical Software, 59(10), 1–23; Harrison, M; Petrou, T. . (2020) Pandas 1.x CookBook. Practical recipes for scientific computing, time series analysis, and exploratory data analysis using Python.

Folha de Pagamentos da Administração Indireta. Dados Abertos, 2019-2021. Disponível em:
<<http://dados.prefeitura.sp.gov.br/dataset/folha-de-pagamentos/>>. Acesso em: 23/11/2021.

Dados Abertos. Portal da transparência. Disponível em
<<http://transparencia.vilavelha.es.gov.br/transparenciaweb/DadosAbertos.Lista.aspx/>>. Acesso em: 23/11/2021.

Folha de Pagamento(Salário dos servidores). Prefeitura Miguel Pereira, portal da transparência. Disponível em:
<<https://transparencia.pmmp.rj.gov.br/?serv=88/>>. Acesso em: 23/11/2021

Tabela 7060. IBGE. Disponível em: <<https://sidra.ibge.gov.br/tabela/7060#resultado/>>. Acesso em: 23/11/2021.

IPCA. Tabela Nacional. Disponível em:
<<https://apisidra.ibge.gov.br/values/t/1737/n1/all/v/all/p/all/d/v63%202,v69%202,v2266%2013,v2263%202,v2264%202,v2265%202?formato=json/>>. Acesso em: 23/11/2021

PIB nacional. Disponível em: <<http://www.ipeadata.gov.br/ExibeSerie.aspx?serid=521274780&module=M/>>. Acesso em: 23/11/2021.



Agradecemos pela atenção!



Contato da Equipe



Débora Azevedo

 deboramendes
azevedo



Igor Cruz

 igorfreitascruz



Rodrigo Silva

 rodrigo-fs



Sabrina Lopes

 sabrinalcosta

