

Portfolio de Aprendizado

Smart
Decision
Consulting

2022

Sumário

| | |
|---|----|
| Acesso aos Dados | 2 |
| Conectando ao ambiente | 2 |
| Definições para uso do SAS Studio | 3 |
| Definições para uso do Jupyter Notebook (Python) | 3 |
| Importando os arquivos no SAS Studio (PROC IMPORT) | 4 |
| Importando os arquivos no Python (PANDAS,PYODBC) | 5 |
| Transformando os dados em tabela..... | 5 |
| Exploração dos dados | 6 |
| Executando o proc contents, proc means, proc freq em todas as tabelas. | 6 |
| Exploração dos Dados..... | 7 |
| Manipulação dos dados | 7 |
| Exportando as tabelas prontas da pasta WORK para MY_LIB/JOAO | 7 |
| Criando um acesso do Python para o SAS | 8 |
| Importando todas as tabelas do MY_LIBRARIES/JOAO | 8 |
| Manipulação da tabela Regiões | 9 |
| Manipulação da tabela VENDAS..... | 9 |
| Join das tabelas | 12 |
| Análise dos dados e Relatórios | 15 |
| Importando os dados no Power Bi | 15 |

Projeto: Transformação digital do processo de acompanhamento do negócio

O desafio nessa etapa do projeto consiste em simular uma demanda realista de um ambiente de trabalho. Simulando uma empresa que acompanha as suas vendas de forma simples, utilizando excel e arquivos em texto. Visando transformar esse processo mais seguro e robusto, a equipe da SDC precisa executar um projeto do início ao fim, importando, limpando e corrigindo os dados recebidos e por fim apresentando através de um dashboard executivo os seus resultados.

Acesso aos Dados

Conectando ao ambiente

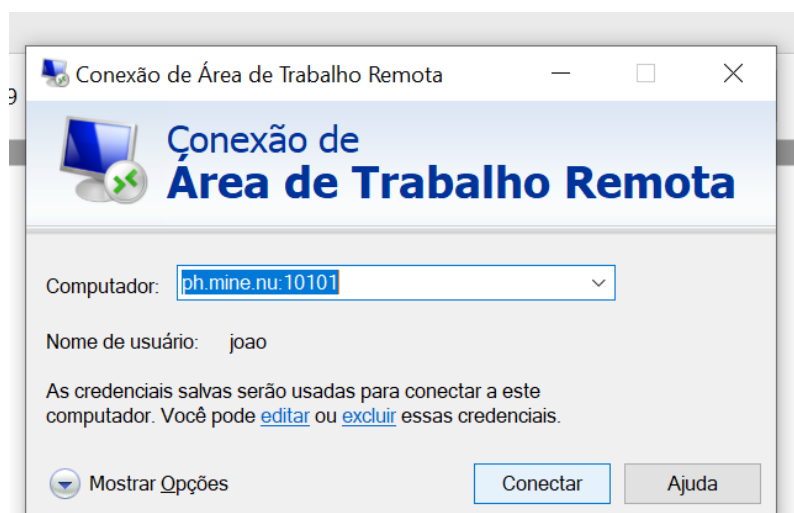
Para acessar a estação de trabalho seguem os dados:

Windows: mstsc(Remote Desktop)

Endereço: xxxxxx.xxxxx.xxxxx

User: joao

Passwd: XXXXXXXXXXXX



Definições para uso do SAS Studio

Efetuar o login e senha

Criar/salvar o(s) programa(s) na pasta **sdccacademy**(/home/joao/sdccacademy)

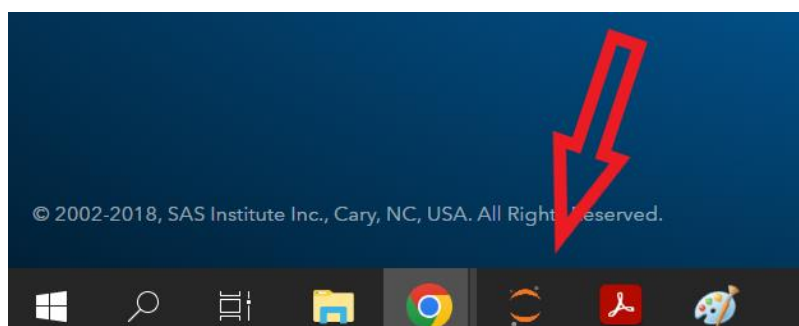
Seus **arquivos** de dados origem estão em /sasdata/sdccacademy/joao/

Utilize a **library** em seu nome(joao) para **armazenar** as **tabelas** criadas e depois tratadas(data cleansing/data quality)



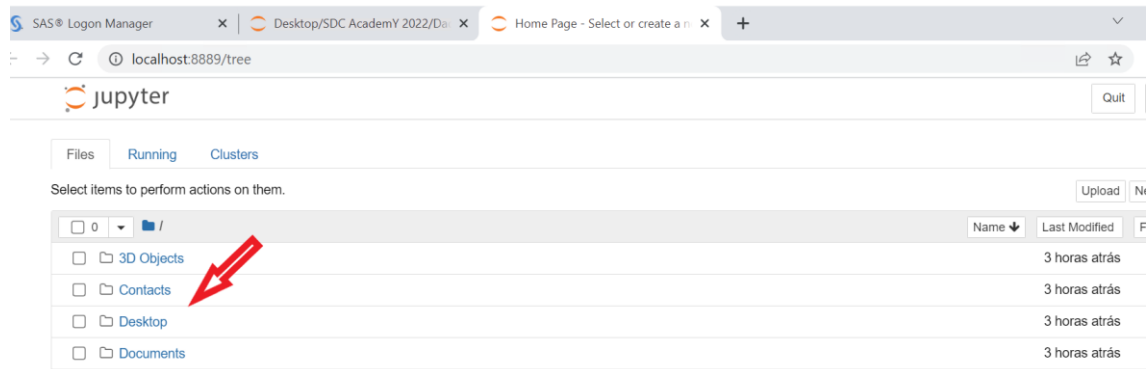
Definições para uso do Jupyter Notebook (Python)

Abra o Jupyter Notebook



Vá até a sua **pasta** em:

C:\Users\joao\Desktop\SDCAcademY2022\Codigo_e_Portfolio



Salvar seu **projeto** com o nome *code_joao.ipynb*

Seus **arquivos** de dados origem estão em *C:\Users\joao\Desktop\SDCAcademY2022\Dados*

Importando os arquivos no SAS Studio (PROC IMPORT)

```
code_joao.sas x
CODE LOG RESULTS
/*IMPORTANDO DOS DADOS PARA PASTA WORK*/
1
2
3
4 proc import out=work.vendas
5     datafile="/sasdata/sdcacademy/joao/DadosACADEMY.xlsx" dbms=xlsx replace;
6     SHEET="Vendas";
7     getnames=YES;
8 run;
9
10 proc import out=work.vendedor
11     datafile="/sasdata/sdcacademy/joao/DadosACADEMY.xlsx" dbms=xlsx replace;
12     SHEET="Vendedor";
13     getnames=YES;
14 run;
15
```

Importando os arquivos no Python (PANDAS,PYODBC)

In [1]: # IMPORTANDO MODULOS PANDAS E PYODBC

```
import pandas as pd
import pyodbc
```

In [2]: #IMPORTANDO TABELA "DadosACADEMY2019.xlsx"

```
dados_ex= pd.read_excel(r"c:\Users\joao\Desktop\SDC AcademY 2022\Dados\DadosACADEMY2019.xlsx")
print(dados_ex)
```

| | CodProduto | CodCor | CodTamanho | CodEstado | DataVenda | Vendedor | \ |
|---|------------|--------|------------|-----------|------------|----------|---|
| 0 | 2909 | 7 | 1 | 22 | 2019-05-02 | 4 | |
| 1 | 1223 | 2 | 2 | 18 | 2019-02-15 | 2 | |
| 2 | 166 | 6 | 3 | 11 | 2019-02-24 | 4 | |
| 3 | 1781 | 4 | 2 | 11 | 2019-02-16 | 2 | |
| 4 | 2671 | 3 | 2 | 24 | 2019-08-24 | 2 | |

In [5]: #IMPORTANDO DADOS "Estados.accdb"

```
conn = pyodbc.connect(r'Driver={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=C:\Users\joao\Desktop\SDC AcademY 2022\Dados\Estados.accdb')
cursor = conn.cursor()
cursor.execute('select * from Estados')

for row in cursor:
    print(row)
```

```
(1,0, 'Acre', 'AC', 'Rio Branco', 0.12, 2.0)
(2,0, 'Alagoas', 'AL', 'Maceió', 0.09, 1.0)
(3,0, 'Amapá', 'AP', 'Macapá', 0.13, 2.0)
```

Transformando os dados em tabela

In [7]: #ALTERANDO TABELA PARA ARQUIVO CSV

```
estados.to_csv('estados.csv', index = False)
```

In [8]: #VISUALIZANDO OS DADOS

```
estados=pd.read_csv('estados.csv')
estados
```

Out[8]:

| | CodEstado | Nome | Sigla | Capital | Perclmposto | CodRegiao |
|---|-----------|---------|-------|------------|-------------|-----------|
| 0 | 1.0 | Acre | AC | Rio Branco | 0.12 | 2.0 |
| 1 | 2.0 | Alagoas | AL | Maceió | 0.09 | 1.0 |
| 2 | 3.0 | Amapá | AP | Macapá | 0.13 | 2.0 |

Exploração dos dados

Executando o proc contents, proc means, proc freq em todas as tabelas.

```
130  
131 /*TABELA VENDEDOR*/  
132  
133 PROC CONTENTS DATA=vendedor;  
134 RUN;  
135  
136 PROC MEANS DATA=vendedor;  
137 RUN;  
138  
139 PROC FREQ DATA=vendedor;  
140 RUN;  
141
```

Conforme visualizado nos resultados dos comandos logo acima, foi detectado algumas incongruências na tabela “VENDAS”.

| Alphabetic List of Variables and Attributes | | | | | | |
|---|-------------|------|-----|-----------|----------|-------------|
| # | Variable | Type | Len | Format | Informat | Label |
| 2 | CodCor | Num | 8 | BEST. | | CodCor |
| 4 | CodEstado | Num | 8 | BEST. | | CodEstado |
| 1 | CodProduto | Num | 8 | BEST. | | CodProduto |
| 3 | CodTamanho | Num | 8 | BEST. | | CodTamanho |
| 5 | DataVenda | Num | 8 | MMDDYY10. | | DataVenda |
| 7 | QtdeVendida | Char | 5 | \$5. | \$5. | QtdeVendida |
| 6 | Vendedor | Char | 1 | \$1. | \$1. | Vendedor |

| | | | | |
|-----------------------|----|------|-------|--------|
| 95 | 74 | 0.49 | 14690 | 97.85 |
| 96 | 88 | 0.59 | 14778 | 98.43 |
| 97 | 78 | 0.52 | 14856 | 98.95 |
| 98 | 76 | 0.51 | 14932 | 99.46 |
| 99 | 81 | 0.54 | 15013 | 100.00 |
| Frequency Missing = 3 | | | | |

Exploração dos Dados

Manipulação dos dados

```
/*DELETANDO OS DADOS MISSING DA TABELA "VENDAS"*/  
  
DATA vendas;  
  set work.vendas;  
  where vendedor not is missing;  
  vendedor=vendedor;  
RUN;  
  
DATA vendas;  
  set work.vendas;  
  where codproduto not is missing;  
  codproduto=codproduto;  
RUN;  
  
/*DELETANDO OS TIPOS "NÃO" NUMERICOS DA VARIALVEL "VENDEDOR"*/  
  
DATA vendas;  
  set work.vendas;  
  
  if vendedor="A" then  
    delete;  
  
  if vendedor="B" then  
    delete;  
  
  if vendedor="C" then  
    delete;  
  
  if vendedor="X" then  
    delete;  
RUN;
```

Exportando as tabelas prontas da pasta WORK para MY_LIB/JOAO

```
proc copy in=work out=joao;  
run;
```


Criando um acesso do Python para o SAS

In [9]: *#CRIANDO UM ACESSO PARA O SAS*

```
import saspy
```

```
sas = saspy.SASsession(cfgname='vmacademy2')
```

Username: joao

Password:

SAS Connection established. Workspace UniqueI

Importando todas as tabelas do MY_LIBRARIES/JOAO

```
cr = sas.sasdata('cores', libref='joao')
gr = sas.sasdata('grupos', libref='joao')
pr = sas.sasdata('produtos', libref='joao')
re = sas.sasdata('regioes', libref='joao')
tm = sas.sasdata('tamanhos', libref='joao')
tt = sas.sasdata('texto_transpose', libref='joao')
vd = sas.sasdata('vendas', libref='joao')
vn = sas.sasdata('vendedor', libref='joao')
```

#CRIANDO AS VARIÁVEIS COM O NOME DAS TABELAS

```
cores=cr.to_df()
grupos=gr.to_df()
produtos=pr.to_df()
regioes=re.to_df()
tamanhos=tm.to_df()
texto=tt.to_df()
vendas=vd.to_df()
vendedor=vn.to_df()
```

Manipulação da tabela Regiões

Foi verificado que na tabela REGIOES não tinha a região SUL.

```
#INCLUINDO A REGIÃO "SUL" NA TABELA REGIOES
```

```
regioes
```

| | CodRegiao | Nome |
|---|-----------|--------------|
| 0 | 1.0 | Nordeste |
| 1 | 2.0 | Norte |
| 2 | 3.0 | Centro Oeste |
| 3 | 4.0 | Sudeste |

```
regioes.loc[4] = [5, "Sul"]
```

Manipulação da tabela VENDAS

```
#TRATANDO A TABELA VENDAS
```

```
vendas
```

| | CodProduto | CodCor | CodTamanho | CodEstado | DataVenda | Vendedor | QtdeVendida |
|-------|------------|--------|------------|-----------|------------|----------|-------------|
| 0 | 1816.0 | 3.0 | 4.0 | 3.0 | 2021-07-11 | 1 | 164 |
| 1 | 582.0 | 2.0 | 1.0 | 10.0 | 2020-01-14 | 5 | 95 |
| 2 | 1519.0 | 7.0 | 2.0 | 19.0 | 2021-04-19 | 3 | 27 |
| 3 | 2949.0 | 7.0 | 1.0 | 24.0 | 2022-05-28 | 2 | 45 |
| 4 | 2528.0 | 7.0 | 1.0 | 11.0 | 2020-04-27 | 5 | 160 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 14984 | 1901.0 | 8.0 | 2.0 | 19.0 | 2020-02-18 | 3 | 20.00 |
| 14985 | 1901.0 | 2.0 | 1.0 | 25.0 | 2022-04-10 | 5 | 6000 |
| 14986 | 1900.0 | 5.0 | 3.0 | 25.0 | 2022-05-01 | 3 | 5000 |
| 14987 | 1901.0 | 8.0 | 4.0 | 19.0 | 2021-12-19 | 4 | 2800 |
| 14988 | 1900.0 | 4.0 | 1.0 | 19.0 | 2021-03-18 | 1 | 3200 |

14989 rows × 7 columns

Logo acima já verificamos que na coluna “qtdevendida” temos alguns dados divergentes.

Aplicando o comando `info()`, conseguimos verificar isso com outros detalhes.

```
vendas.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14989 entries, 0 to 14988
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   CodProduto      14989 non-null  float64
1   CodCor           14989 non-null  float64
2   CodTamanho      14989 non-null  float64
3   CodEstado       14989 non-null  float64
4   DataVenda       14989 non-null  datetime64[ns]
5   Vendedor        14989 non-null  object
6   QtdeVendida     14989 non-null  object ←
dtypes: datetime64[ns](1), float64(4), object(2)
memory usage: 819.8+ KB
```

```
#SELECIONANDO A VARIÁVEL "QTDEVENDIDA"
```

```
vendas['QtdeVendida']
```

```
0      164
1      95
2      27
3      45
4     160
...
14984  20.00 ←
14985   6000
14986   5000
14987   2800
14988   3200
Name: QtdeVendida, Length: 14989, dtype: object
```

```
#ENCONTRANDO NA VARIÁVEL O OBJETO DIVERGENTE
```

```
for ponto in vendas['QtdeVendida']:
    if '.' in ponto:
        print(ponto)
        break
```

```
20.00
```

Pra corrigir esse dado, foi necessário alterar o tipo da coluna de STRING para FLOAT.

```
#ALTERANDO O TIPO DA COLUNA "QTDEVENDIDA" DE STRING PARA FLOAT  
vendas['QtdeVendida'] = vendas['QtdeVendida'].astype(float)
```

Numa base de quase 15000 vendas, tínhamos 4 linhas muito discrepantes das outras.

| | |
|-------|------|
| 14985 | 6000 |
| 14986 | 5000 |
| 14987 | 2800 |
| 14988 | 3200 |

Nesse caso foi realizada a alteração desses dados.

```
vendas['QtdeVendida'] > 1000  
  
0      False  
1      False  
2      False  
3      False  
4      False  
...  
14984   False  
14985    True  
14986    True  
14987    True  
14988    True  
Name: QtdeVendida, Length: 14989, dtype: bool
```

```
vendas.loc[14985, 'QtdeVendida']  
  
6000.0
```

```
vendas.loc[14985, 'QtdeVendida'] = "60.0"
```

```
vendas.loc[14986, 'QtdeVendida']
```

```
5000.0
```

```
vendas.loc[14986, 'QtdeVendida']="50.0"
```

```
vendas.loc[14987, 'QtdeVendida']
```

```
2800.0
```

```
vendas.loc[14987, 'QtdeVendida']="28.0"
```

```
vendas.loc[14988, 'QtdeVendida']
```

```
3200.0
```

```
vendas.loc[14988, 'QtdeVendida']="32.0"
```

Já na coluna “vendedor” foi realizada a alteração do tipo da coluna.

```
##ALTERANDO O TIPO DA COLUNA "VENDEDOR" DE STRING PARA FLOAT
```

```
vendas['Vendedor'] = vendas['Vendedor'].astype(float)
```

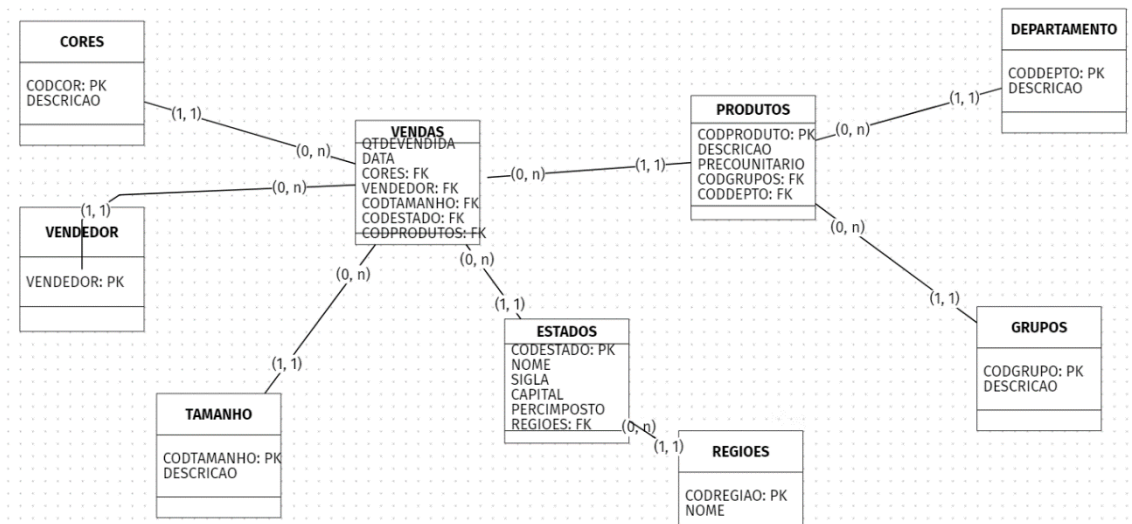
Com o processo aparentemente finalizado de exploração e preparação dos dados, foi realizada a junção das tabelas.

Join das tabelas

As primeiras tabelas foram as dos dados de 2019 à 2022.

```
vendas = pd.concat([dados_2019, vendas])
```

Pra auxiliar a junção das tabelas usamos o app.brmodeloweb.com.



Visto que a tabela “Produtos” não ligava a chave CodDeppto, foi constatado que existia uma outra tabela que não foi enviada no início do projeto.

Solicitado a área de negócios essa nova tabela, deu-se novamente o processo de exploração e preparação dos dados.

#IMPORTANDO A TABELA DEPARTAMENTO

```
departamento=pd.read_csv(r"c:\Users\joao\Desktop\SDC AcademY 2022\Dados\Departamentos.csv")
departamento
```

| CodDeppto;Descricao | |
|---------------------|-------------------------|
| 0 | 1;Artigos Esportivos |
| 1 | 2;Casamentos |
| 2 | 3;Roupas |
| 3 | 4;Infantil |
| 4 | 5;Ferramentas |
| 5 | 6;Utensílios Domésticos |
| 6 | 7;Alimentos |
| 7 | 8;Automotivos |
| 8 | 9;Jardinagem |
| 9 | 10;Náutica |

```
#TRATANDO A TABELA DEPARTAMENTO
```

```
departamento[['CodDepto', 'Descricao']] = departamento['CodDepto;Descricao'].str.split(';', expand=True)
```

```
departamento=departamento.drop(columns=['CodDepto;Descricao'])
```

```
departamento.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   CodDepto    10 non-null    object
1   Descricao   10 non-null    object
dtypes: object(2)
memory usage: 288.0+ bytes
```

```
departamento['CodDepto'] = departamento['CodDepto'].astype(float)
```

```
departamento=departamento.rename(columns={'Descricao': 'Departamento'})
```

Ainda preparando a tabela final, foram realizadas algumas alterações em nomes e tipos de colunas e a inclusão de uma nova coluna com o nome “Faturamento”.

```
tabela['CodCor'] = tabela['CodCor'].astype(int)
tabela['CodRegiao'] = tabela['CodRegiao'].astype(int)
tabela['CodEstado'] = tabela['CodEstado'].astype(int)
tabela['CodTamanho'] = tabela['CodTamanho'].astype(int)
tabela['CodProduto'] = tabela['CodProduto'].astype(int)
tabela['Vendedor'] = tabela['Vendedor'].astype(int)
tabela['CodGrupo'] = tabela['CodGrupo'].astype(int)
tabela['CodDepto'] = tabela['CodDepto'].astype(int)
```

```
tabela['QtdeVendida'] = tabela['QtdeVendida'].astype(float)
```

```
tabela['Imposto'] = tabela['PrecoUnitario'] * tabela['PercImposto']
```

```
tabela['Imposto'] = tabela['Imposto'].apply(lambda x: "{:,.2f}".format(x))
```

```
tabela['Imposto'] = tabela['Imposto'].astype(str)
```

```
tabela['Faturamento'] = tabela['PrecoUnitario'] * tabela['QtdeVendida']
```

```
tabela['QtdeVendida'] = tabela['QtdeVendida'].astype(int)
```

Por fim, foi realizado a criação do arquivo CSV e a exportação da tabela para o diretório "DADOS".

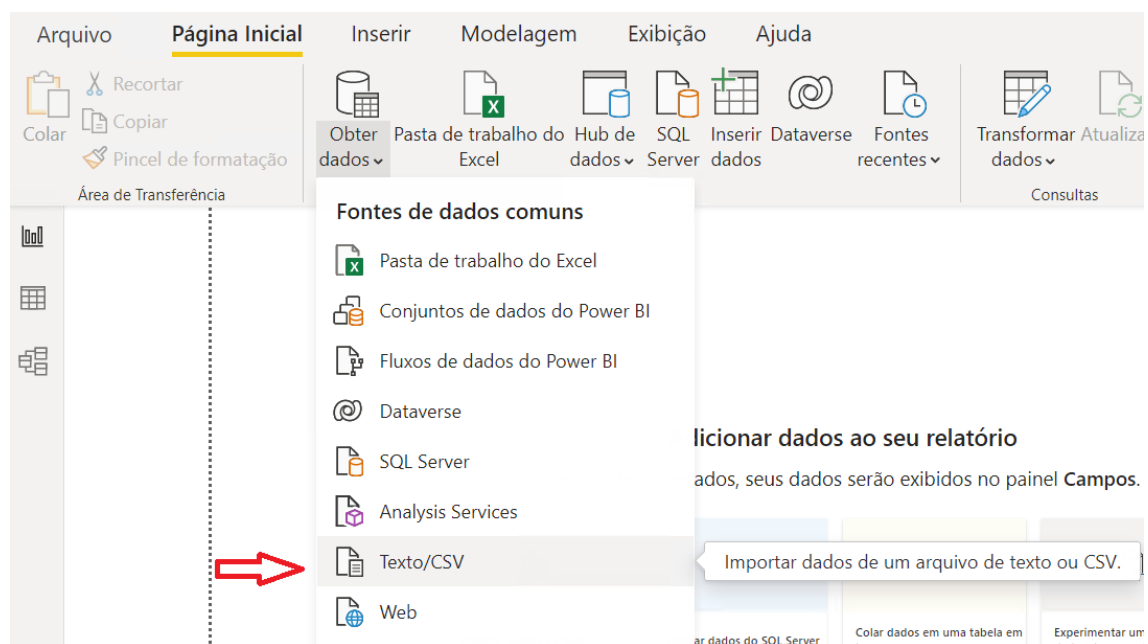
```
#EXPORTANDO TABELA PARA O DIRETÓRIO "DADOS"
```

```
diretorio = 'C:\\Users\\joao\\Desktop\\SDC AcademY 2022\\Dados'
```

```
tabela.to_csv(diretorio + 'tabela_final_2022.csv', index=False)
```

Análise dos dados e Relatórios

Importando os dados no Power Bi

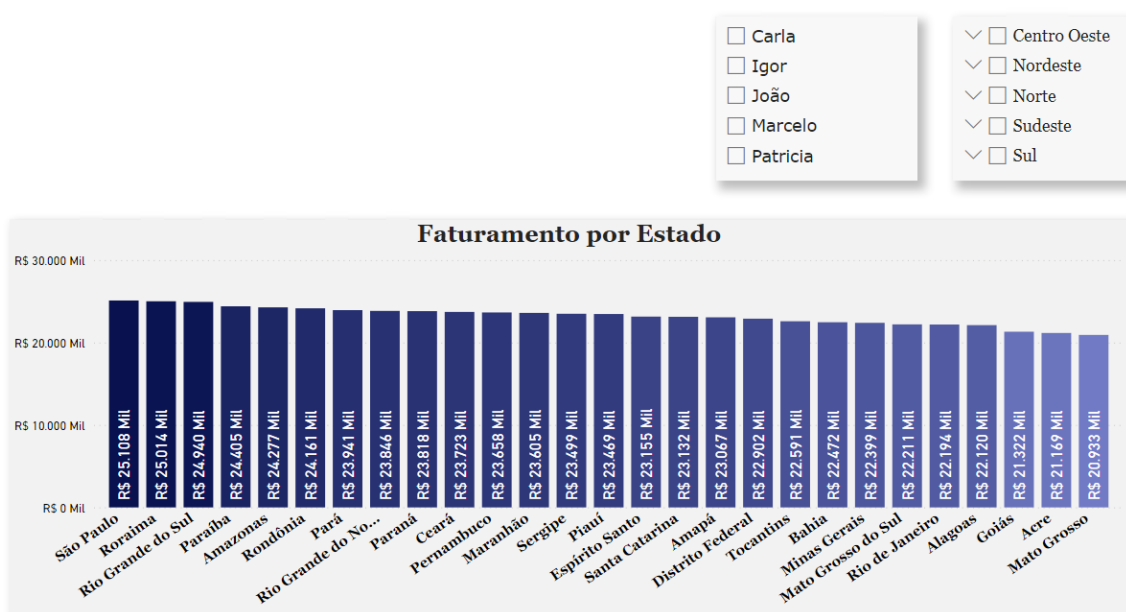


Solicitado via e-mail as visões do negócio.

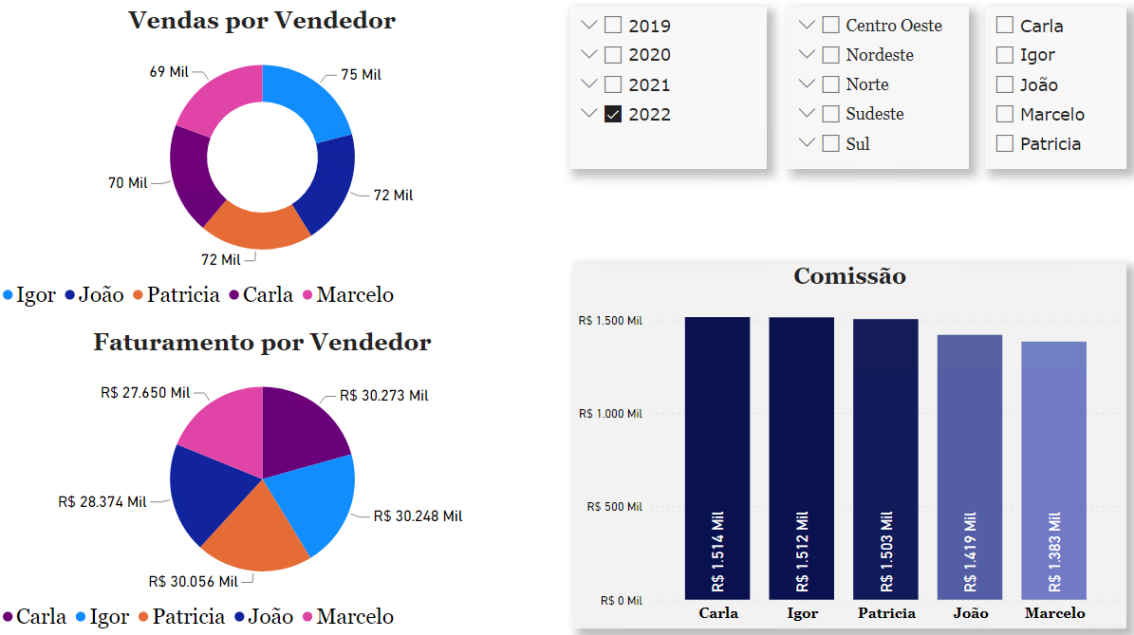
- Gerar um relatório no PowerBI com as seguintes informações:
 - Vendas: Produto/Período/Estado
 - Comissão: Vendedor/Período/Região
 - Disponibilizar filtros por: mês, ano, região
 - Visões Estratégicas
 - Qual o Estado que gera maior faturamento?
 - Quais os melhores vendedores em cada região?
 - Qual o produto mais vendido?
 - Quanto foi pago de imposto por estado? Usar recurso de mapa.

De acordo com a demanda, foi criado algumas colunas como “Imposto” e “Comissão” e editado as nomenclaturas monetárias de algumas colunas para melhor visualização e compreensão dos dashboards.

A primeira visão solicitada foi do estado com o maior faturamento.



A segunda foi quais são os melhores vendedores por região.



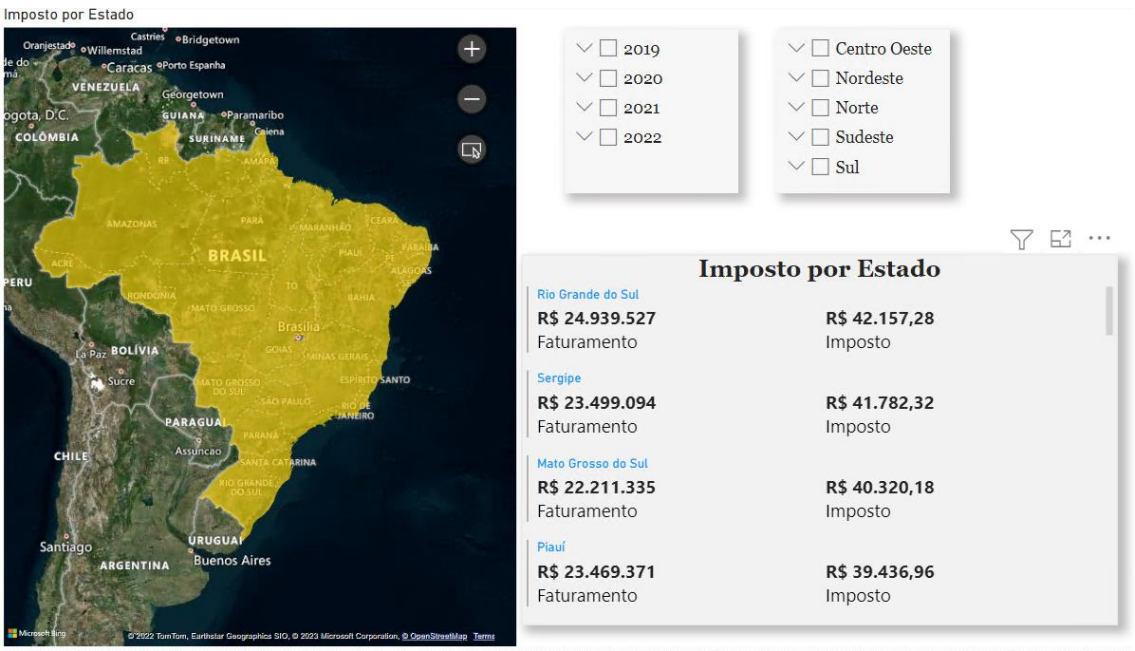
A terceira foi de qual o produto mais vendido.

2019
2020
2021
2022

| QtdeVendida | CodProduto | Departamento | Grupo | PrecoUnitario | Faturamento |
|-------------|------------|-----------------------|----------|---------------|-------------|
| 1665 | 1055 | Jardinagem | Grupo 6 | R\$ 1.510 | R\$ 353.340 |
| 1489 | 853 | Utensílios Domésticos | Grupo 5 | R\$ 1.030 | R\$ 401.803 |
| 1452 | 2675 | Artigos Esportivos | Grupo 9 | R\$ 7.620 | R\$ 362.330 |
| 1447 | 2333 | Infantil | Nacional | R\$ 7.330 | R\$ 478.134 |
| 1437 | 2398 | Artigos Esportivos | Nacional | R\$ 3.290 | R\$ 534.954 |
| 1404 | 1742 | Jardinagem | Nacional | R\$ 1.200 | R\$ 367.200 |
| 1400 | 1649 | Casamentos | Grupo 12 | R\$ 5.240 | R\$ 661.916 |
| 1397 | 51 | Roupas | Grupo 14 | R\$ 4.130 | R\$ 576.961 |
| 1372 | 675 | Automotivos | Grupo 9 | R\$ 5.600 | R\$ 474.488 |
| 1366 | 1890 | Alimentos | Grupo 12 | R\$ 2.930 | R\$ 431.882 |
| 1359 | 456 | Jardinagem | Grupo 12 | R\$ 2.190 | R\$ 354.780 |
| 1344 | 1403 | Jardinagem | Nacional | R\$ 5.410 | R\$ 444.213 |
| 1342 | 424 | Náutica | Grupo 12 | R\$ 7.060 | R\$ 272.019 |
| 1341 | 471 | Alimentos | Grupo 15 | R\$ 5.000 | R\$ 670.500 |

Selegonar tudo
Carla
Igor
João
Marcelo
Patricia

E a última foi de quanto foi pago de imposto por Estado e usar o recurso de mapa.



Finalizando o desafio, foi criada uma cópia de todos os arquivos para o repositório GIT da SDCacademy.