Atvidade Indivdual

Aluno: Carlos Bahia

Prof.: Igor Gondim

1. Nível Infra

```
# bibliotecas para usar com pandas e colab
!pip install gcsfs
!pip install fsspec
```

```
# carregando as bibliotecas
import os
import pandas as pd
import numpy  as np
from google.cloud import storage
from google.colab import drive
drive.mount('/content/drive')

# configuração da chave de segurança
serviceAccount = '/content/drive/MyDrive/central-point-349020-ea14a01a5b63.json'
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = serviceAccount

# acessa o bucket para fazer o download do arquivo
client = storage.Client()

# variável "bucket" recebe o nome da bucket do Cloud Storage
bucket = client.get_bucket('atividade-individual')
```

```
bucket.blob('marketing_campaign.csv')

path = 'gs://atividade-individual/original/marketing_campaign.csv'
df = pd.read_csv(path,sep=';')
print(df)
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_re
         ID  Year_Birth   Education Marital_Status   Income  Kidhome  \
0      5524        1957  Graduation         Single  58138.0        0
1      2174        1954  Graduation         Single  46344.0        1
2      4141        1965  Graduation       Together  71613.0        0
3      6182        1984  Graduation       Together  26646.0        1
4      5324        1981         PhD        Married  58293.0        1
...     ...         ...         ...            ...      ...      ...
2235  10870        1967  Graduation        Married  61223.0        0
2236   4001        1946         PhD       Together  64014.0        2
2237   7270        1981  Graduation       Divorced  56981.0        0
2238   8235        1956      Master       Together  69245.0        0
2239   9405        1954         PhD        Married  52869.0        1

      Teenhome Dt_Customer  Recency  MntWines  ...  NumWebVisitsMonth  \
0            0  2012-09-04       58       635  ...                  7
1            1  2014-03-08       38        11  ...                  5
2            0  2013-08-21       26       426  ...                  4
3            0  2014-02-10       26        11  ...                  6
4            0  2014-01-19       94       173  ...                  5
...        ...         ...      ...       ...  ...                ...
2235         1  2013-06-13       46       709  ...                  5
2236         1  2014-06-10       56       406  ...                  7
2237         0  2014-01-25       91       908  ...                  6
2238         1  2014-01-24        8       428  ...                  3
2239         1  2012-10-15       40        84  ...                  7

      AcceptedCmp3  AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  \
0                0             0             0             0             0
1                0             0             0             0             0
2                0             0             0             0             0
3                0             0             0             0             0
4                0             0             0             0             0
...            ...           ...           ...           ...           ...
```

```
2235            0            0            0            0            0
2236            0            0            0            1            0
2237            0            1            0            0            0
2238            0            0            0            0            0
2239            0            0            0            0            0

      Complain  Z_CostContact  Z_Revenue  Response
0            0              3         11         1
1            0              3         11         0
2            0              3         11         0
3            0              3         11         0
4            0              3         11         0
...        ...            ...        ...       ...
2235         0              3         11         0
2236         0              3         11         0
2237         0              3         11         0
2238         0              3         11         0
2239         0              3         11         1

[2240 rows x 29 columns]
```

2. Nível Pandas

O arquivo está em outra linguagem e deve ter seus dados traduzidos para Português-BR

Realizar a extração corretamente para um dataframe

```
# traduzindo o nome das colunas
df.rename(columns = {'ID':'id', 'Year_Birth':'ano_nascimento', 'Education':'educacao', 'Marital_Status':'estado_civil', 'I
df.rename(columns = {'Dt_Customer':'data_cliente', 'Recency':'tempo_decisao_compra', 'MntWines':'qtde_vinhos_vend_mes', 'M
df.rename(columns = {'MntFishProducts':'qtde_pescados_vend_mes', 'MntSweetProducts':'qtde_doceria_vend_mes', 'MntGoldProds
df.rename(columns = {'NumWebPurchases':'tot_compras_web','NumCatalogPurchases':'tot_compras_catalogo','NumStorePurchases':
df.rename(columns = {'AcceptedCmp4':'adesao_camp_mkt_4','AcceptedCmp5':'adesao_camp_mkt_5','AcceptedCmp1':'adesao_camp_mkt
```

```
df.rename(columns = {'Z_CostContact':'custo_contact_z','Z_Revenue':'receita_z','Response':'resposta'}, inplace = True)



# analisando os conteúdos das colunas que podem ser traduzidos
#print(df)

# as colunas que podem ter seus conteúdos traduzidos são: educação e estado civil
# educação
item_counts = df["educacao"].value_counts()
print(item_counts)
```

```
Graduation    1127
PhD            486
Master         370
2n Cycle       203
Basic           54
Name: educacao, dtype: int64
```

```
# foram identificados na coluna 'educacao' os seguintes conteúdos:

Graduation    1127
PhD            486
Master         370
2n Cycle       203
Basic           54


# gravando a tradução da coluna 'educacao'

df["educacao"].replace({"Graduation": "Bacharel", "PhD": "Doutor(a)", "Master": "Mestrado", "2n Cycle": "Mestrado", "Basic
item_counts = df["educacao"].value_counts()
print(item_counts)
print(df)

# https://www2.ed.gov/about/offices/list/ous/international/usnei/us/edlite-structure-us.html

# 2n Cycle
```

```
# https://www.google.com/search?q=us+education+2n+Cycle&client=firefox-b-d&sxsrf=ALiCzsZn1CfTVx4vH8UZUegRb5zEcbozEg%3A1653

# Basic
# https://www.google.com/search?q=us+basic+education&client=firefox-b-d&sxsrf=ALiCzsbxfHDmlWx1ABTH77XxoyCDchHYvg%3A165377
```

```
Bacharel     1127
Mestrado      573
Doutor(a)     486
Primario       54
Name: educacao, dtype: int64
         id  ano_nascimento   educacao estado_civil  renda_familiar  \
0      5524            1957   Bacharel       Single         58138.0
1      2174            1954   Bacharel       Single         46344.0
2      4141            1965   Bacharel     Together         71613.0
3      6182            1984   Bacharel     Together         26646.0
4      5324            1981  Doutor(a)      Married         58293.0
...     ...             ...        ...          ...             ...
2235  10870            1967   Bacharel      Married         61223.0
2236   4001            1946  Doutor(a)     Together         64014.0
2237   7270            1981   Bacharel     Divorced         56981.0
2238   8235            1956   Mestrado     Together         69245.0
2239   9405            1954  Doutor(a)      Married         52869.0

      qtde_criancas  qtde_adolescentes data_cliente  tempo_decisao_compra  \
0                 0                  0   2012-09-04                    58
1                 1                  1   2014-03-08                    38
2                 0                  0   2013-08-21                    26
3                 1                  0   2014-02-10                    26
4                 1                  0   2014-01-19                    94
...             ...                ...          ...                   ...
2235              0                  1   2013-06-13                    46
2236              2                  1   2014-06-10                    56
2237              0                  0   2014-01-25                    91
2238              0                  1   2014-01-24                     8
2239              1                  1   2012-10-15                    40

      qtde_vinhos_vend_mes  ...  tot_visitas_web_mes  adesao_camp_mkt_3  \
0                      635  ...                    7                  0
1                       11  ...                    5                  0
```

```
2                    426  ...                   4                  0
3                     11  ...                   6                  0
4                    173  ...                   5                  0
...                  ...  ...                 ...                ...
2235                 709  ...                   5                  0
2236                 406  ...                   7                  0
2237                 908  ...                   6                  0
2238                 428  ...                   3                  0
2239                  84  ...                   7                  0

      adesao_camp_mkt_4  adesao_camp_mkt_5  adesao_camp_mkt_1  \
0                     0                  0                  0
1                     0                  0                  0
2                     0                  0                  0
3                     0                  0                  0
4                     0                  0                  0
...                 ...                ...                ...
2235                  0                  0                  0
2236                  0                  0                  1
2237                  1                  0                  0
2238                  0                  0                  0
2239                  0                  0                  0

      adesao_camp_mkt_2  reclamacao  custo_contact_z  receita_z  resposta
```

```
# as colunas que podem ter seus conteúdos traduzidos são: educação e estado civil
# estado civil
item_counts = df["estado_civil"].value_counts()
print(item_counts)
```

```
Married     864
Together    580
Single      480
Divorced    232
Widow        77
Alone         3
Absurd        2
YOLO          2
Name: estado_civil, dtype: int64
```

```
df["estado_civil"].replace({"Married": "casado(a)", "Together": "morando junto", "Single": "solteiro(a)", "Divorced": "div
item_counts = df["estado_civil"].value_counts()
print(item_counts)
print(df)
```

```
casado(a)        864
morando junto    580
solteiro(a)      483
divorciado(a)    232
viúvo(a)          77
Absurd             2
YOLO               2
Name: estado_civil, dtype: int64
         id  ano_nascimento   educacao   estado_civil  renda_familiar  \
0      5524            1957   Bacharel    solteiro(a)         58138.0
1      2174            1954   Bacharel    solteiro(a)         46344.0
2      4141            1965   Bacharel  morando junto         71613.0
3      6182            1984   Bacharel  morando junto         26646.0
4      5324            1981  Doutor(a)      casado(a)         58293.0
...     ...             ...        ...            ...             ...
2235  10870            1967   Bacharel      casado(a)         61223.0
2236   4001            1946  Doutor(a)  morando junto         64014.0
2237   7270            1981   Bacharel  divorciado(a)         56981.0
2238   8235            1956   Mestrado  morando junto         69245.0
2239   9405            1954  Doutor(a)      casado(a)         52869.0

      qtde_criancas  qtde_adolescentes data_cliente  tempo_decisao_compra  \
0                 0                  0   2012-09-04                    58
1                 1                  1   2014-03-08                    38
2                 0                  0   2013-08-21                    26
3                 1                  0   2014-02-10                    26
4                 1                  0   2014-01-19                    94
...             ...                ...          ...                   ...
2235              0                  1   2013-06-13                    46
2236              2                  1   2014-06-10                    56
2237              0                  0   2014-01-25                    91
2238              0                  1   2014-01-24                     8
2239              1                  1   2012-10-15                    40

      qtde_vinhos_vend_mes  ...  tot_visitas_web_mes  adesao_camp_mkt_3  \
0                      635  ...                    7                  0
```

```
1                    11  ...                   5                  0
2                   426  ...                   4                  0
3                    11  ...                   6                  0
4                   173  ...                   5                  0
...                 ...  ...                 ...                ...
2235                709  ...                   5                  0
2236                406  ...                   7                  0
2237                908  ...                   6                  0
2238                428  ...                   3                  0
2239                 84  ...                   7                  0

      adesao_camp_mkt_4  adesao_camp_mkt_5  adesao_camp_mkt_1  \
0                     0                  0                  0
1                     0                  0                  0
2                     0                  0                  0
3                     0                  0                  0
4                     0                  0                  0
...                 ...                ...                ...
2235                  0                  0                  0
2236                  0                  0                  1
2237                  1                  0                  0
2238                  0                  0                  0
```

```
# na coluna "estado_civil" foram encontradas 4 registros com as seguintes classificações:
# Absurd             2
# YOLO               2

# esses 4 registros são despreziveis pois representam 0,4% do dataframe


df.drop(df[df.estado_civil == 'Absurd' ].index, inplace=True)
df.drop(df[df.estado_civil == 'YOLO'   ].index, inplace=True)

item_counts = df["estado_civil"].value_counts()
print(item_counts)
print(df)
```

```
casado(a)        864
morando junto    580
solteiro(a)      483
```

```
solteiro(a)      485
divorciado(a)    232
viúvo(a)          77
Name: estado_civil, dtype: int64
         id  ano_nascimento   educacao   estado_civil  renda_familiar  \
0      5524            1957   Bacharel    solteiro(a)         58138.0
1      2174            1954   Bacharel    solteiro(a)         46344.0
2      4141            1965   Bacharel  morando junto         71613.0
3      6182            1984   Bacharel  morando junto         26646.0
4      5324            1981  Doutor(a)      casado(a)         58293.0
...     ...             ...        ...            ...             ...
2235  10870            1967   Bacharel      casado(a)         61223.0
2236   4001            1946  Doutor(a)  morando junto         64014.0
2237   7270            1981   Bacharel  divorciado(a)         56981.0
2238   8235            1956   Mestrado  morando junto         69245.0
2239   9405            1954  Doutor(a)      casado(a)         52869.0

      qtde_criancas  qtde_adolescentes data_cliente  tempo_decisao_compra  \
0                 0                  0   2012-09-04                    58
1                 1                  1   2014-03-08                    38
2                 0                  0   2013-08-21                    26
3                 1                  0   2014-02-10                    26
4                 1                  0   2014-01-19                    94
...             ...                ...          ...                   ...
2235              0                  1   2013-06-13                    46
2236              2                  1   2014-06-10                    56
2237              0                  0   2014-01-25                    91
2238              0                  1   2014-01-24                     8
2239              1                  1   2012-10-15                    40

      qtde_vinhos_vend_mes  ...  tot_visitas_web_mes  adesao_camp_mkt_3  \
0                      635  ...                    7                  0
1                       11  ...                    5                  0
2                      426  ...                    4                  0
3                       11  ...                    6                  0
4                      173  ...                    5                  0
...                    ...  ...                  ...                ...
2235                   709  ...                    5                  0
2236                   406  ...                    7                  0
2237                   908  ...                    6                  0
2238                   428  ...                    3                  0
2239                    84  ...                    7                  0
```

```
      adesao_camp_mkt_4  adesao_camp_mkt_5  adesao_camp_mkt_1  \
0                     0                  0                  0
1                     0                  0                  0
2                     0                  0                  0
3                     0                  0                  0
4                     0                  0                  0
...                 ...                ...                ...
2235                  0                  0                  0
2236                  0                  0                  1
2237                  1                  0                  0
2238                  0                  0                  0
2239                  0                  0                  0
```

```
# Realizar o drop(se necessário) de colunas do dataframe realizando o comentário do porque da exclusão
# NÃO houve drop de colunas

# somei a quantidade de NaNs em todo dataframe e encontrei 24 registros
df.isna().sum().sum()
```

```
24
```

```
# identifiquei as colunas que contém os NaNs em todo dataframe
for coluna in df.columns:
    print( coluna, '  Qtde de NaNs '   ,df[coluna].isna().sum())
```

```
id   Qtde de NaNs  0
ano_nascimento   Qtde de NaNs  0
educacao   Qtde de NaNs  0
estado_civil   Qtde de NaNs  0
renda_familiar   Qtde de NaNs  24
qtde_criancas   Qtde de NaNs  0
qtde_adolescentes   Qtde de NaNs  0
data_cliente   Qtde de NaNs  0
tempo_decisao_compra   Qtde de NaNs  0
qtde_vinhos_vend_mes   Qtde de NaNs  0
qtde_frutas_vend_mes   Qtde de NaNs  0
qtde_carnes_vend_mes   Qtde de NaNs  0
```

```
qtde_pescados_vend_mes   Qtde de NaNs  0
qtde_doceria_vend_mes   Qtde de NaNs  0
qtde_artigo_ouro_vend_mes   Qtde de NaNs  0
tot_negocios_fechados   Qtde de NaNs  0
tot_compras_web   Qtde de NaNs  0
tot_compras_catalogo   Qtde de NaNs  0
tot_compras_lojas   Qtde de NaNs  0
tot_visitas_web_mes   Qtde de NaNs  0
adesao_camp_mkt_3   Qtde de NaNs  0
adesao_camp_mkt_4   Qtde de NaNs  0
adesao_camp_mkt_5   Qtde de NaNs  0
adesao_camp_mkt_1   Qtde de NaNs  0
adesao_camp_mkt_2   Qtde de NaNs  0
reclamacao   Qtde de NaNs  0
custo_contact_z   Qtde de NaNs  0
receita_z   Qtde de NaNs  0
resposta   Qtde de NaNs  0
```

```
# foram encontrados 24 rows com NaN de um total de 2241 rows do dataframe
# esses 24 rows representam  1,07% de toda base, esse valor está dentro da margem aceitavel para se dropada

df.dropna(inplace=True)
for coluna in df.columns:
    print( coluna, '  Qtde de NaNs '   ,df[coluna].isna().sum())
```

```
id   Qtde de NaNs  0
ano_nascimento   Qtde de NaNs  0
educacao   Qtde de NaNs  0
estado_civil   Qtde de NaNs  0
renda_familiar   Qtde de NaNs  0
qtde_criancas   Qtde de NaNs  0
qtde_adolescentes   Qtde de NaNs  0
data_cliente   Qtde de NaNs  0
tempo_decisao_compra   Qtde de NaNs  0
qtde_vinhos_vend_mes   Qtde de NaNs  0
qtde_frutas_vend_mes   Qtde de NaNs  0
qtde_carnes_vend_mes   Qtde de NaNs  0
qtde_pescados_vend_mes   Qtde de NaNs  0
qtde_doceria_vend_mes   Qtde de NaNs  0
```

```
qtde_artigo_ouro_vend_mes   Qtde de NaNs  0
tot_negocios_fechados   Qtde de NaNs  0
tot_compras_web   Qtde de NaNs  0
tot_compras_catalogo   Qtde de NaNs  0
tot_compras_lojas   Qtde de NaNs  0
tot_visitas_web_mes   Qtde de NaNs  0
adesao_camp_mkt_3   Qtde de NaNs  0
adesao_camp_mkt_4   Qtde de NaNs  0
adesao_camp_mkt_5   Qtde de NaNs  0
adesao_camp_mkt_1   Qtde de NaNs  0
adesao_camp_mkt_2   Qtde de NaNs  0
reclamacao   Qtde de NaNs  0
custo_contact_z   Qtde de NaNs  0
receita_z   Qtde de NaNs  0
resposta   Qtde de NaNs  0
```

3. Nivel PySpark

```
# instalando PySpark e algumas bibliotecas
#!pip install pyspark
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType,StructField, StringType, IntegerType, FloatType, DoubleType, DateType
from pyspark.sql.functions import *
from pyspark     import SparkConf
import pyspark.sql.functions as F


# 3.1 - fazendo a conexão com a Sparksession

spark = (SparkSession.builder
                    .master("local")
                    .appName("atividade-individual")
                    .config('spark.ui.port', '4050')
                    .getOrCreate()
        )
```

```
# já que o df está tratado no pandas posso converter para um df em PySpark
spark.conf.set("spark.sql.execution.arrow.enabled","true")
dfspark = spark.createDataFrame(df)
```

```
# visualizando a conversão de pandas para pyspark
dfspark.show(5)
dfspark.printSchema()
dfspark.count()
```

```
+----+--------------+---------+-------------+--------------+-------------+-----------------+------------+-----------
|  id|ano_nascimento| educacao| estado_civil|renda_familiar|qtde_criancas|qtde_adolescentes|data_cliente|tempo_decis
+----+--------------+---------+-------------+--------------+-------------+-----------------+------------+-----------
|5524|          1957| Bacharel|  solteiro(a)|       58138.0|            0|                0|  2012-09-04|
|2174|          1954| Bacharel|  solteiro(a)|       46344.0|            1|                1|  2014-03-08|
|4141|          1965| Bacharel|morando junto|       71613.0|            0|                0|  2013-08-21|
|6182|          1984| Bacharel|morando junto|       26646.0|            1|                0|  2014-02-10|
|5324|          1981|Doutor(a)|    casado(a)|       58293.0|            1|                0|  2014-01-19|
+----+--------------+---------+-------------+--------------+-------------+-----------------+------------+-----------
only showing top 5 rows

root
 |-- id: long (nullable = true)
 |-- ano_nascimento: long (nullable = true)
 |-- educacao: string (nullable = true)
 |-- estado_civil: string (nullable = true)
 |-- renda_familiar: double (nullable = true)
 |-- qtde_criancas: long (nullable = true)
 |-- qtde_adolescentes: long (nullable = true)
 |-- data_cliente: string (nullable = true)
 |-- tempo_decisao_compra: long (nullable = true)
 |-- qtde_vinhos_vend_mes: long (nullable = true)
 |-- qtde_frutas_vend_mes: long (nullable = true)
 |-- qtde_carnes_vend_mes: long (nullable = true)
 |-- qtde_pescados_vend_mes: long (nullable = true)
 |-- qtde_doceria_vend_mes: long (nullable = true)
 |-- qtde_antigo_ouro_vend_mes: long (nullable = true)
```

```
 |-- qtde_artigo_ouro_vend_mes: long (nullable = true)
 |-- tot_negocios_fechados: long (nullable = true)
 |-- tot_compras_web: long (nullable = true)
 |-- tot_compras_catalogo: long (nullable = true)
 |-- tot_compras_lojas: long (nullable = true)
 |-- tot_visitas_web_mes: long (nullable = true)
 |-- adesao_camp_mkt_3: long (nullable = true)
 |-- adesao_camp_mkt_4: long (nullable = true)
 |-- adesao_camp_mkt_5: long (nullable = true)
 |-- adesao_camp_mkt_1: long (nullable = true)
 |-- adesao_camp_mkt_2: long (nullable = true)
 |-- reclamacao: long (nullable = true)
 |-- custo_contact_z: long (nullable = true)
 |-- receita_z: long (nullable = true)
 |-- resposta: long (nullable = true)

2212
```

```
# 3.2 - usando a função "cast" para alterar o StructType do dataframe

# double usa 64 bits e float usa 32 bits      # long usa 64 bits e int usa 32 bits

dfspark_2 = dfspark \
  .withColumn("id"                        , dfspark["id"].cast(StringType()))   \
  .withColumn("ano_nascimento"            , dfspark["ano_nascimento"].cast(StringType()))   \
  .withColumn("renda_familiar"            , dfspark["renda_familiar"].cast(FloatType()))  \
  .withColumn("qtde_criancas"             , dfspark["qtde_criancas"].cast(IntegerType()))  \
  .withColumn("qtde_adolescentes"         , dfspark["qtde_adolescentes"].cast(IntegerType()))   \
  .withColumn("data_cliente"              , dfspark["data_cliente"].cast(DateType()))   \
  .withColumn("tempo_decisao_compra"      , dfspark["tempo_decisao_compra"].cast(IntegerType()))   \
  .withColumn("qtde_vinhos_vend_mes"      , dfspark["qtde_vinhos_vend_mes"].cast(IntegerType()))   \
  .withColumn("qtde_frutas_vend_mes"      , dfspark["qtde_frutas_vend_mes"].cast(IntegerType()))   \
  .withColumn("qtde_carnes_vend_mes"      , dfspark["qtde_carnes_vend_mes"].cast(IntegerType()))   \
  .withColumn("qtde_pescados_vend_mes"    , dfspark["qtde_pescados_vend_mes"].cast(IntegerType()))   \
  .withColumn("qtde_doceria_vend_mes"     , dfspark["qtde_doceria_vend_mes"].cast(IntegerType()))   \
  .withColumn("qtde_artigo_ouro_vend_mes" , dfspark["qtde_artigo_ouro_vend_mes"].cast(IntegerType()))   \
  .withColumn("tot_negocios_fechados"     , dfspark["tot_negocios_fechados"].cast(IntegerType()))   \
  .withColumn("tot_compras_web"           , dfspark["tot_compras_web"].cast(IntegerType()))   \
  .withColumn("tot_compras_catalogo"      , dfspark["tot_compras_catalogo"].cast(IntegerType()))   \
```

```
  .withColumn("tot_compras_lojas"         , dfspark["tot_compras_lojas"].cast(IntegerType()))   \
  .withColumn("tot_visitas_web_mes"       , dfspark["tot_visitas_web_mes"].cast(IntegerType()))   \
  .withColumn("adesao_camp_mkt_1"         , dfspark["adesao_camp_mkt_1"].cast(IntegerType()))   \
  .withColumn("adesao_camp_mkt_2"         , dfspark["adesao_camp_mkt_2"].cast(IntegerType()))   \
  .withColumn("adesao_camp_mkt_3"         , dfspark["adesao_camp_mkt_3"].cast(IntegerType()))   \
  .withColumn("adesao_camp_mkt_4"         , dfspark["adesao_camp_mkt_4"].cast(IntegerType()))   \
  .withColumn("adesao_camp_mkt_5"         , dfspark["adesao_camp_mkt_5"].cast(IntegerType()))   \
  .withColumn("reclamacao"                , dfspark["reclamacao"].cast(StringType()))   \
  .withColumn("custo_contact_z"           , dfspark["custo_contact_z"].cast(DoubleType()))   \
  .withColumn("receita_z"                 , dfspark["receita_z"].cast(DoubleType()))   \
  .withColumn("resposta"                  , dfspark["resposta"].cast(IntegerType()))   \
```

```
# verificando as alterações de StructureType
dfspark_2.printSchema()
```

```
root
 |-- id: string (nullable = true)
 |-- ano_nascimento: string (nullable = true)
 |-- educacao: string (nullable = true)
 |-- estado_civil: string (nullable = true)
 |-- renda_familiar: float (nullable = true)
 |-- qtde_criancas: integer (nullable = true)
 |-- qtde_adolescentes: integer (nullable = true)
 |-- data_cliente: date (nullable = true)
 |-- tempo_decisao_compra: integer (nullable = true)
 |-- qtde_vinhos_vend_mes: integer (nullable = true)
 |-- qtde_frutas_vend_mes: integer (nullable = true)
 |-- qtde_carnes_vend_mes: integer (nullable = true)
 |-- qtde_pescados_vend_mes: integer (nullable = true)
 |-- qtde_doceria_vend_mes: integer (nullable = true)
 |-- qtde_artigo_ouro_vend_mes: integer (nullable = true)
 |-- tot_negocios_fechados: integer (nullable = true)
 |-- tot_compras_web: integer (nullable = true)
 |-- tot_compras_catalogo: integer (nullable = true)
 |-- tot_compras_lojas: integer (nullable = true)
 |-- tot_visitas_web_mes: integer (nullable = true)
 |-- adesao_camp_mkt_3: integer (nullable = true)
 |-- adesao_camp_mkt_4: integer (nullable = true)
```

```
 |-- adesao_camp_mkt_5: integer (nullable = true)
 |-- adesao_camp_mkt_1: integer (nullable = true)
 |-- adesao_camp_mkt_2: integer (nullable = true)
 |-- reclamacao: string (nullable = true)
 |-- custo_contact_z: double (nullable = true)
 |-- receita_z: double (nullable = true)
 |-- resposta: integer (nullable = true)
```

```
# Pré-análise de dados

dfspark_2.summary().show()
```

```
+-------+------------------+------------------+--------+------------+------------------+------------------+---------
|summary|                id|    ano_nascimento|educacao|estado_civil|    renda_familiar|     qtde_criancas| qtde_ado
+-------+------------------+------------------+--------+------------+------------------+------------------+---------
|  count|              2212|              2212|    2212|        2212|              2212|              2212|
|   mean|  5587.73191681736|1968.8110307414104|    null|        null| 52232.51084990959|0.4425858951175407|0.5054249
| stddev|3247.9441278218037|11.982065047000503|    null|        null|25187.455359009906|0.5370524479338292|0.5442578
|    min|                 0|              1893|Bacharel|   casado(a)|            1730.0|                 0|
|    25%|            2814.0|            1959.0|    null|        null|           35196.0|                 0|
|    50%|            5455.0|            1970.0|    null|        null|           51373.0|                 0|
|    75%|            8420.0|            1977.0|    null|        null|           68487.0|                 1|
|    max|              9999|              1996|Primario|    viúvo(a)|          666666.0|                 2|
+-------+------------------+------------------+--------+------------+------------------+------------------+---------
```

```
# 3.4 - Realizar a mudança de nome de pelo menos 2 colunas

dfspark_2 = dfspark_2.withColumnRenamed("id","id_cliente").withColumnRenamed("ano_nascimento", "ano_de_nascimento")
dfspark_2.show(5)
```

```
+----------+-----------------+---------+-------------+--------------+-------------+-----------------+------------+--
|id_cliente|ano_de_nascimento| educacao| estado_civil|renda_familiar|qtde_criancas|qtde_adolescentes|data_cliente|te
+----------+-----------------+---------+-------------+--------------+-------------+-----------------+------------+--
|      5524|             1957| Bacharel|  solteiro(a)|       58138.0|            0|                0|  2012-09-04|
|      2174|             1954| Bacharel|  solteiro(a)|       46344.0|            1|                1|  2014-03-08|
|      4141|             1965| Bacharel|morando junto|       71613.0|            0|                0|  2013-08-21|
|      6182|             1984| Bacharel|morando junto|       26646.0|            1|                0|  2014-02-10|
```

```
|      5324|             1981|Doutor(a)|    casado(a)|       58293.0|            1|                0|  2014-01-19|
+----------+-----------------+---------+-------------+--------------+-------------+-----------------+------------+--
only showing top 5 rows
```

```
# 3.5 - Deverá criar pelo menos duas novas colunas contendo alguma informação relevante sobre as outras colunas já exister

# familias sem  filhos, sem criancas ou adolescentes

# criação da coluna total de filhos (qtde_criancas + qtde_adolescentes)
dfspark_2 = dfspark_2.withColumn('tot_de_filhos', F.col('qtde_criancas') + F.col('qtde_adolescentes'))

# criação da coluna número total de adesões de todas as campanhas
dfspark_2 = dfspark_2.withColumn('tot_adesao_camp_mkt', F.col('adesao_camp_mkt_1') +  F.col('adesao_camp_mkt_2') + F.col('
```

```
+----------+-----------------+---------+-------------+--------------+-------------+-----------------+------------+--
|id_cliente|ano_de_nascimento| educacao| estado_civil|renda_familiar|qtde_criancas|qtde_adolescentes|data_cliente|te
+----------+-----------------+---------+-------------+--------------+-------------+-----------------+------------+--
|      5524|             1957| Bacharel|  solteiro(a)|       58138.0|            0|                0|  2012-09-04|
|      2174|             1954| Bacharel|  solteiro(a)|       46344.0|            1|                1|  2014-03-08|
|      4141|             1965| Bacharel|morando junto|       71613.0|            0|                0|  2013-08-21|
|      6182|             1984| Bacharel|morando junto|       26646.0|            1|                0|  2014-02-10|
|      5324|             1981|Doutor(a)|    casado(a)|       58293.0|            1|                0|  2014-01-19|
+----------+-----------------+---------+-------------+--------------+-------------+-----------------+------------+--
only showing top 5 rows
```

```
# agregação, contando o número de cliente por ano de nascimento

dfspark_2
dfspark_2.groupBy("ano_de_nascimento").count().show(n=10)
```

```
+-----------------+-----+
|ano_de_nascimento|count|
```

```
+-----------------+-----+
|             1953|   35|
|             1957|   40|
|             1987|   27|
|             1956|   55|
|             1958|   52|
|             1943|    6|
|             1972|   78|
|             1988|   29|
|             1977|   52|
|             1971|   86|
+-----------------+-----+
only showing top 10 rows
```

```
# 3.6 - ordenacao
print('------------ ordenacao ascendente ------------')
dfspark_2.sort(F.col('ano_de_nascimento').asc()).show()
print('------------ ordenacao descendente ------------')
dfspark_2.sort(F.col('ano_de_nascimento').desc()).show()
```

```
------------ ordenacao ascendente ------------
+----------+-----------------+---------+-------------+--------------+-------------+-----------------+------------+--
|id_cliente|ano_de_nascimento| educacao| estado_civil|renda_familiar|qtde_criancas|qtde_adolescentes|data_cliente|te
+----------+-----------------+---------+-------------+--------------+-------------+-----------------+------------+--
|     11004|             1893| Mestrado|  solteiro(a)|       60182.0|            0|                1|  2014-05-17|
|      1150|             1899|Doutor(a)|morando junto|       83532.0|            0|                0|  2013-09-26|
|      7829|             1900| Mestrado|divorciado(a)|       36640.0|            1|                0|  2013-09-26|
|      6663|             1940|Doutor(a)|  solteiro(a)|       51141.0|            0|                0|  2013-07-08|
|      6932|             1941|Doutor(a)|    casado(a)|       93027.0|            0|                0|  2013-04-13|
|      2968|             1943|Doutor(a)|divorciado(a)|       48948.0|            0|                0|  2013-02-01|
|      7106|             1943|Doutor(a)|    casado(a)|       75865.0|            0|                0|  2014-03-31|
|      4994|             1943| Mestrado|  solteiro(a)|       77598.0|            0|                0|  2013-10-01|
|      6142|             1943| Mestrado|    casado(a)|       65073.0|            0|                0|  2013-08-20|
|      8800|             1943|Doutor(a)|divorciado(a)|       48948.0|            0|                0|  2013-02-01|
|      1453|             1943|Doutor(a)|     viúvo(a)|       57513.0|            0|                0|  2013-07-06|
|      3711|             1944| Mestrado|    casado(a)|       80184.0|            0|                0|  2014-03-01|
|      6605|             1944|Doutor(a)|divorciado(a)|       55614.0|            0|                0|  2013-11-27|
|      4587|             1944| Mestrado|     viúvo(a)|       45006.0|            0|                0|  2013-07-18|
|      1740|             1944| Bacharel|    casado(a)|       55956.0|            0|                0|  2014-04-07|
```

```
|       466|             1944| Bacharel|    casado(a)|       65275.0|           0|                0|  2013-04-03|
|      9930|             1944|Doutor(a)|  solteiro(a)|       82716.0|           0|                0|  2013-11-05|
|      4310|             1944| Bacharel|    casado(a)|       80589.0|           0|                0|  2014-01-22|
|      9370|             1945|Doutor(a)|    casado(a)|       65846.0|           0|                0|  2013-05-17|
|      9260|             1945|Doutor(a)|    casado(a)|       70356.0|           0|                0|  2012-11-05|
+----------+-----------------+---------+-------------+--------------+------------+-----------------+------------+--
only showing top 20 rows

------------ ordenacao descendente ------------
+----------+-----------------+--------+-------------+--------------+------------+-----------------+------------+---
|id_cliente|ano_de_nascimento|educacao| estado_civil|renda_familiar|qtde_criancas|qtde_adolescentes|data_cliente|tem
+----------+-----------------+--------+-------------+--------------+------------+-----------------+------------+---
|      9909|             1996|Mestrado|    casado(a)|        7500.0|           0|                0|  2012-11-09|
|       193|             1996|Primario|    casado(a)|       14421.0|           0|                0|  2014-02-17|
|      4427|             1995|Mestrado|  solteiro(a)|       83257.0|           0|                0|  2012-09-18|
|      3661|             1995|Mestrado|  solteiro(a)|       80617.0|           0|                0|  2012-10-12|
|      8315|             1995|Bacharel|  solteiro(a)|       34824.0|           0|                0|  2014-03-26|
|     10548|             1995|Bacharel|  solteiro(a)|       71163.0|           0|                0|  2014-03-09|
|      5184|             1995|Bacharel|  solteiro(a)|       34824.0|           0|                0|  2014-03-26|
|      5067|             1994|Bacharel|morando junto|       80134.0|           0|                0|  2014-02-14|
|      6905|             1994|Bacharel|morando junto|       80685.0|           0|                0|  2012-08-22|
|     10619|             1994|Bacharel|  solteiro(a)|       95529.0|           0|                0|  2012-12-03|
|      2669|             1993|Bacharel|  solteiro(a)|       74293.0|           0|                0|  2014-05-04|
|     10037|             1993|Bacharel|  solteiro(a)|       74293.0|           0|                0|  2014-05-04|
|      4483|             1993|Bacharel|  solteiro(a)|       72354.0|           0|                0|  2013-04-17|
|      5080|             1993|Bacharel|  solteiro(a)|       70515.0|           0|                0|  2013-10-21|
|       821|             1992|Mestrado|  solteiro(a)|       92859.0|           0|                0|  2012-10-19|
|      1379|             1992|Mestrado|morando junto|       42670.0|           0|                0|  2013-04-27|
|      3386|             1992|Bacharel|    casado(a)|       34935.0|           0|                0|  2013-06-21|
|      3005|             1992|Bacharel|  solteiro(a)|       83528.0|           0|                0|  2014-05-01|
|      8560|             1992|Bacharel|  solteiro(a)|       48789.0|           0|                0|  2012-09-10|
|      4136|             1992|Primario|  solteiro(a)|        7500.0|           1|                0|  2012-12-03|
+----------+-----------------+--------+-------------+--------------+------------+-----------------+------------+---
only showing top 20 rows
```

```
# 3.7 - usando windows function

from pyspark.sql.window import Window
from pyspark.sql.functions import row_number
windowSpec  = Window.partitionBy("educacao").orderBy("ano_de_nascimento")
```

```
windowSpec  = Window.partitionBy("educacao").orderBy("ano_de_nascimento")

dfspark_2.withColumn("row_number",row_number().over(windowSpec)).show(truncate=False)
```

```
+----------+-----------------+--------+-------------+--------------+-------------+-----------------+------------+---
|id_cliente|ano_de_nascimento|educacao|estado_civil |renda_familiar|qtde_criancas|qtde_adolescentes|data_cliente|tem
+----------+-----------------+--------+-------------+--------------+-------------+-----------------+------------+---
|466       |1944             |Bacharel|casado(a)    |65275.0       |0            |0                |2013-04-03  |9
|1740      |1944             |Bacharel|casado(a)    |55956.0       |0            |0                |2014-04-07  |22
|4310      |1944             |Bacharel|casado(a)    |80589.0       |0            |0                |2014-01-22  |25
|819       |1945             |Bacharel|viúvo(a)     |63285.0       |0            |0                |2013-11-05  |84
|10711     |1945             |Bacharel|casado(a)    |69755.0       |0            |0                |2013-10-02  |23
|4939      |1946             |Bacharel|morando junto|37760.0       |0            |0                |2012-08-31  |20
|8652      |1946             |Bacharel|morando junto|37760.0       |0            |0                |2012-08-31  |20
|5029      |1946             |Bacharel|casado(a)    |18100.0       |0            |0                |2013-08-06  |14
|1577      |1946             |Bacharel|casado(a)    |78569.0       |0            |0                |2014-02-19  |14
|1553      |1946             |Bacharel|morando junto|82657.0       |0            |0                |2013-09-27  |71
|6963      |1947             |Bacharel|solteiro(a)  |77457.0       |0            |0                |2014-03-05  |85
|10814     |1947             |Bacharel|morando junto|70321.0       |0            |0                |2013-01-16  |6
|6203      |1947             |Bacharel|morando junto|74485.0       |0            |0                |2013-08-24  |58
|7990      |1947             |Bacharel|casado(a)    |27469.0       |0            |0                |2012-08-02  |2
|4137      |1948             |Bacharel|morando junto|70666.0       |0            |0                |2013-12-06  |29
|10486     |1948             |Bacharel|casado(a)    |77142.0       |0            |0                |2013-05-29  |54
|5147      |1948             |Bacharel|solteiro(a)  |90842.0       |0            |0                |2013-07-29  |57
|10909     |1948             |Bacharel|casado(a)    |92344.0       |0            |0                |2014-01-15  |9
|10341     |1948             |Bacharel|morando junto|51315.0       |0            |0                |2014-02-23  |45
|6103      |1948             |Bacharel|casado(a)    |42192.0       |0            |0                |2013-09-06  |40
+----------+-----------------+--------+-------------+--------------+-------------+-----------------+------------+---
only showing top 20 rows
```

```
# usando windows function

from pyspark.sql.window import Window
from pyspark.sql.functions import row_number
windowSpec  = Window.partitionBy("tot_compras_web").orderBy("tot_visitas_web_mes")

dfspark_2.withColumn("web_compras_por_visitas",row_number().over(windowSpec)).show(truncate=False)
```

```
+----------+-----------------+---------+-------------+--------------+-------------+-----------------+------------+--
```

```
|id_cliente|ano_de_nascimento|educacao |estado_civil |renda_familiar|qtde_criancas|qtde_adolescentes|data_cliente|te
+----------+-----------------+---------+-------------+--------------+-------------+-----------------+------------+--
|8475      |1973             |Doutor(a)|casado(a)    |157243.0      |0            |1                |2014-03-01  |98
|5555      |1975             |Bacharel |divorciado(a)|153924.0      |0            |0                |2014-02-07  |81
|1501      |1982             |Doutor(a)|casado(a)    |160803.0      |0            |0                |2012-08-04  |21
|11181     |1949             |Doutor(a)|casado(a)    |156924.0      |0            |0                |2013-08-29  |85
|5376      |1979             |Bacharel |casado(a)    |2447.0        |1            |0                |2013-01-06  |42
|1503      |1976             |Doutor(a)|morando junto|162397.0      |1            |1                |2013-06-03  |31
|4931      |1977             |Bacharel |morando junto|157146.0      |0            |0                |2013-04-29  |13
|10492     |1959             |Bacharel |morando junto|38285.0       |2            |1                |2014-06-24  |96
|10401     |1976             |Mestrado |morando junto|26326.0       |0            |0                |2013-09-04  |12
|2724      |1981             |Mestrado |solteiro(a)  |36143.0       |1            |0                |2014-03-30  |33
|7196      |1950             |Doutor(a)|casado(a)    |41145.0       |1            |1                |2014-02-08  |20
|7788      |1983             |Doutor(a)|casado(a)    |23536.0       |1            |0                |2014-06-04  |53
|6679      |1966             |Bacharel |solteiro(a)  |33279.0       |0            |0                |2014-06-12  |29
|9283      |1978             |Bacharel |solteiro(a)  |60199.0       |1            |2                |2013-09-12  |49
|2166      |1960             |Mestrado |viúvo(a)     |46779.0       |1            |1                |2013-07-12  |55
|9523      |1982             |Bacharel |morando junto|40479.0       |1            |0                |2013-08-17  |95
|8977      |1985             |Primario |solteiro(a)  |16581.0       |0            |0                |2013-01-12  |51
|10141     |1960             |Mestrado |divorciado(a)|39228.0       |0            |0                |2013-05-10  |1
|7901      |1971             |Mestrado |casado(a)    |34109.0       |0            |1                |2013-11-06  |39
|8080      |1986             |Bacharel |solteiro(a)  |26816.0       |0            |0                |2012-08-17  |50
+----------+-----------------+---------+-------------+--------------+-------------+-----------------+------------+--
only showing top 20 rows
```

```
dfspark_2.show(5)
```

```
+----------+-----------------+---------+-------------+--------------+-------------+-----------------+------------+--
|id_cliente|ano_de_nascimento| educacao| estado_civil|renda_familiar|qtde_criancas|qtde_adolescentes|data_cliente|te
+----------+-----------------+---------+-------------+--------------+-------------+-----------------+------------+--
|      5524|             1957| Bacharel|  solteiro(a)|       58138.0|            0|                0|  2012-09-04|
|      2174|             1954| Bacharel|  solteiro(a)|       46344.0|            1|                1|  2014-03-08|
|      4141|             1965| Bacharel|morando junto|       71613.0|            0|                0|  2013-08-21|
|      6182|             1984| Bacharel|morando junto|       26646.0|            1|                0|  2014-02-10|
|      5324|             1981|Doutor(a)|    casado(a)|       58293.0|            1|                0|  2014-01-19|
+----------+-----------------+---------+-------------+--------------+-------------+-----------------+------------+--
only showing top 5 rows
```

```
# 4 - Nivel SparkSQL

dfspark_2.createOrReplaceTempView("MARKETING_CAMPAIGN")
df2 = spark.sql("SELECT * from MARKETING_CAMPAIGN")
df2.printSchema()
df2.show()
```

```
root
 |-- id: string (nullable = true)
 |-- ano_nascimento: string (nullable = true)
 |-- educacao: string (nullable = true)
 |-- estado_civil: string (nullable = true)
 |-- renda_familiar: float (nullable = true)
 |-- qtde_criancas: integer (nullable = true)
 |-- qtde_adolescentes: integer (nullable = true)
 |-- data_cliente: date (nullable = true)
 |-- tempo_decisao_compra: integer (nullable = true)
 |-- qtde_vinhos_vend_mes: integer (nullable = true)
 |-- qtde_frutas_vend_mes: integer (nullable = true)
 |-- qtde_carnes_vend_mes: integer (nullable = true)
 |-- qtde_pescados_vend_mes: integer (nullable = true)
 |-- qtde_doceria_vend_mes: integer (nullable = true)
 |-- qtde_artigo_ouro_vend_mes: integer (nullable = true)
 |-- tot_negocios_fechados: integer (nullable = true)
 |-- tot_compras_web: integer (nullable = true)
 |-- tot_compras_catalogo: integer (nullable = true)
 |-- tot_compras_lojas: integer (nullable = true)
 |-- tot_visitas_web_mes: integer (nullable = true)
 |-- adesao_camp_mkt_3: integer (nullable = true)
 |-- adesao_camp_mkt_4: integer (nullable = true)
 |-- adesao_camp_mkt_5: integer (nullable = true)
 |-- adesao_camp_mkt_1: integer (nullable = true)
 |-- adesao_camp_mkt_2: integer (nullable = true)
 |-- reclamacao: string (nullable = true)
 |-- custo_contact_z: double (nullable = true)
 |-- receita_z: double (nullable = true)
 |-- resposta: integer (nullable = true)
```

```
+----+--------------+---------+-------------+--------------+-------------+-----------------+------------+-----------
|  id|ano_nascimento| educacao| estado_civil|renda_familiar|qtde_criancas|qtde_adolescentes|data_cliente|tempo_decis
+----+--------------+---------+-------------+--------------+-------------+-----------------+------------+-----------
```

```
|5524|          1957| Bacharel|  solteiro(a)|       58138.0|            0|                0|  2012-09-04|
|2174|          1954| Bacharel|  solteiro(a)|       46344.0|            1|                1|  2014-03-08|
|4141|          1965| Bacharel|morando junto|       71613.0|            0|                0|  2013-08-21|
|6182|          1984| Bacharel|morando junto|       26646.0|            1|                0|  2014-02-10|
|5324|          1981|Doutor(a)|    casado(a)|       58293.0|            1|                0|  2014-01-19|
|7446|          1967| Mestrado|morando junto|       62513.0|            0|                1|  2013-09-09|
| 965|          1971| Bacharel|divorciado(a)|       55635.0|            0|                1|  2012-11-13|
|6177|          1985|Doutor(a)|    casado(a)|       33454.0|            1|                0|  2013-05-08|
|4855|          1974|Doutor(a)|morando junto|       30351.0|            1|                0|  2013-06-06|
|5899|          1950|Doutor(a)|morando junto|        5648.0|            1|                1|  2014-03-13|
| 387|          1976| Primario|    casado(a)|        7500.0|            0|                0|  2012-11-13|
|2125|          1959| Bacharel|divorciado(a)|       63033.0|            0|                0|  2013-11-15|
|8180|          1952| Mestrado|divorciado(a)|       59354.0|            1|                1|  2013-11-15|
|2569|          1987| Bacharel|    casado(a)|       17323.0|            0|                0|  2012-10-10|
|2114|          1946|Doutor(a)|  solteiro(a)|       82800.0|            0|                0|  2012-11-24|
|9736|          1980| Bacharel|    casado(a)|       41850.0|            1|                1|  2012-12-24|
|4939|          1946| Bacharel|morando junto|       37760.0|            0|                0|  2012-08-31|
|6565|          1949| Mestrado|    casado(a)|       76995.0|            0|                1|  2013-03-28|
|2278|          1985| Mestrado|  solteiro(a)|       33812.0|            1|                0|  2012-11-03|
|9360|          1982| Bacharel|    casado(a)|       37040.0|            0|                0|  2012-08-08|
+----+--------------+---------+-------------+--------------+-------------+-----------------+------------+-----------
only showing top 20 rows
```

```
# 4.1 - analisando a renda familiar minima por grau de instrução do individuo
df2 = spark.sql("SELECT educacao, min(renda_familiar) from MARKETING_CAMPAIGN  group by educacao  ")
df2.show()
```

```
+---------+-------------------+
| educacao|min(renda_familiar)|
+---------+-------------------+
| Primario|             7500.0|
|Doutor(a)|             4023.0|
| Bacharel|             1730.0|
| Mestrado|             6560.0|
+---------+-------------------+
```

```
# 4.2 - analisando a renda familiar máxima por grau de instrução do individuo
```

```
df2 = spark.sql("SELECT educacao, max(renda_familiar) from MARKETING_CAMPAIGN  group by educacao  ")
df2.show()
```

```
+---------+------------------+
| educacao|max(renda_familiar)|
+---------+------------------+
| Primario|           34445.0|
|Doutor(a)|          162397.0|
| Bacharel|          666666.0|
| Mestrado|          157733.0|
+---------+------------------+
```

```
# 4.3 - analisando a renda familiar média por grau de instrução do individuo
df2 = spark.sql("SELECT educacao, avg(renda_familiar) from MARKETING_CAMPAIGN  group by educacao  ")
df2.show()
```

```
+---------+------------------+
| educacao|avg(renda_familiar)|
+---------+------------------+
| Primario|  20306.25925925926|
|Doutor(a)|  56177.51983298539|
| Bacharel|  52696.58565022422|
| Mestrado| 51021.367021276594|
+---------+------------------+
```

```
# 4.4 - analisando a renda familiar minima pelo estado civil do individuo
df2 = spark.sql("SELECT estado_civil, min(renda_familiar) from MARKETING_CAMPAIGN  group by estado_civil  ")
df2.show()
```

```
+-------------+------------------+
| estado_civil|min(renda_familiar)|
+-------------+------------------+
|     viúvo(a)|           22123.0|
|divorciado(a)|            1730.0|
|  solteiro(a)|            3502.0|
|    casado(a)|            2447.0|
|morando junto|            5648.0|
```

```
+-------------+------------------+
```

```
# 4.5 - analisando a renda familiar máxima pelo estado civil do individuo
df2 = spark.sql("SELECT estado_civil, max(renda_familiar) from MARKETING_CAMPAIGN  group by estado_civil  ")
df2.show()
```

```
+-------------+------------------+
| estado_civil|max(renda_familiar)|
+-------------+------------------+
|     viúvo(a)|           85620.0|
|divorciado(a)|          153924.0|
|  solteiro(a)|          113734.0|
|    casado(a)|          160803.0|
|morando junto|          666666.0|
+-------------+------------------+
```

```
# 4.6··analisando·a·renda·familiar·média·pelo·estado·civil·do·individuo
df2 = spark.sql("SELECT estado_civil, avg(renda_familiar) from MARKETING_CAMPAIGN  group by estado_civil  ")
df2.show()
```

```
+-------------+------------------+
| estado_civil|avg(renda_familiar)|
+-------------+------------------+
|     viúvo(a)|  56481.55263157895|
|divorciado(a)|  52834.22844827586|
|  solteiro(a)| 50949.740506329115|
|    casado(a)|  51724.97899649942|
|morando junto|  53245.53403141361|
+-------------+------------------+
```

```
# 5 - salvando os dataframes no Google Cloud Storage

# salvando para csv
dfspark_2.write.format("csv").save("marketing_campaign_tratado")
```

```
# configuração da chave de segurança
serviceAccount = '/content/drive/MyDrive/central-point-349020-ea14a01a5b63.json'
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = serviceAccount


#CÓDIGO QUE ACESSA A BUCKET CRIADA E FAZ O DOWNLOAD DOS ARQUIVOS VIA PANDAS
client = storage.Client()

#CRIAR UMA VARIÁVEL CHAMADA BUCKET QUE VAI RECEBER O NOME DA BUCKET DO CLOUD STORAGE
bucket = client.get_bucket('atividade-individual')
#USAR O MÉTODO BLOB PARA RETORNAR O NOME DO ARQUIVO (JSON, CSV, PARQUET)
bucket.blob('marketing_campaign_tratado.csv')

bucket.blob('tratado/marketing_campaign_tratado.csv').upload_from_string(df.to_csv(), 'text/csv')
```

```
# 6···inserindo·arquivo·tratado·no·MongoDB

import csv
import pandas as pd
from modules.connector_mongodb import connector
from pymongo import MongoClient
import pymongo
from bson.json_util import dumps, loads



connection_DB= pymongo.MongoClient(f"mongodb+srv://atividadeindividual:atividadeindividual@cluster0.xweww.mongodb.net/ativ

print("teste de conexão: ")
print(connection_DB)
```

```
print("teste de leitura do arquivo csv")
df_dados = pd.read_csv("C:\\Users\\Dalva\\Downloads\\Soulcode\\Projeto Individual\\marketing_campaign.csv", sep  =';')
print(df_dados)
# Converte de DF para Dicionário
data_dict = df_dados.to_dict(orient='records')

# Conecta ao banco de dados
db = connection_DB.marketing_campaign

#  insere dados no MongoDB Atlas
db.marketing_campaign.insert_many(data_dict)

print('-------------------------------------------------------------')
print('Dados inseridos com sucesso  !!')
print(db)
print('-----------
```

Inicializacao do DF para quando der erro no Colab

```
#!pip install gcsfs     # para usar com pandas e colab
#!pip install fsspec


import os
import pandas as pd
import numpy  as np
from google.cloud import storage
from google.colab import drive
drive.mount('/content/drive')
```

```
# configuração da chave de segurança
serviceAccount = '/content/drive/MyDrive/central-point-349020-ea14a01a5b63.json'
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = serviceAccount


#CÓDIGO QUE ACESSA A BUCKET CRIADA E FAZ O DOWNLOAD DOS ARQUIVOS VIA PANDAS
client = storage.Client()

#CRIAR UMA VARIÁVEL CHAMADA BUCKET QUE VAI RECEBER O NOME DA BUCKET DO CLOUD STORAGE
bucket = client.get_bucket('atividade-individual')
#USAR O MÉTODO BLOB PARA RETORNAR O NOME DO ARQUIVO (JSON, CSV, PARQUET)
bucket.blob('marketing_campaign.csv')

path = 'gs://atividade-individual/original/marketing_campaign.csv'
df = pd.read_csv(path,sep=';')
print(df)



# traduzindo o nome das colunas
df.rename(columns = {'ID':'id', 'Year_Birth':'ano_nascimento', 'Education':'educacao', 'Marital_Status':'estado_civil', 'I
df.rename(columns = {'Dt_Customer':'data_cliente', 'Recency':'tempo_decisao_compra', 'MntWines':'qtde_vinhos_vend_mes', 'M
df.rename(columns = {'MntFishProducts':'qtde_pescados_vend_mes', 'MntSweetProducts':'qtde_doceria_vend_mes', 'MntGoldProds
df.rename(columns = {'NumWebPurchases':'tot_compras_web','NumCatalogPurchases':'tot_compras_catalogo','NumStorePurchases':
df.rename(columns = {'AcceptedCmp4':'adesao_camp_mkt_4','AcceptedCmp5':'adesao_camp_mkt_5','AcceptedCmp1':'adesao_camp_mkt
df.rename(columns = {'Z_CostContact':'custo_contact_z','Z_Revenue':'receita_z','Response':'resposta'}, inplace = True)


# analisando os conteúdos das colunas que podem ser traduzidos
#print(df)

# as colunas que podem ter seus conteúdos traduzidos são: educação e estado civil
# educação
item_counts = df["educacao"].value_counts()
print(item_counts)


# gravando a tradução da coluna 'educacao'
```

```
# gravando a tradução da coluna "educacao"

df["educacao"].replace({"Graduation": "Bacharel", "PhD": "Doutor(a)", "Master": "Mestrado", "2n Cycle": "Mestrado", "Basic
item_counts = df["educacao"].value_counts()
print(item_counts)
print(df)


df["estado_civil"].replace({"Married": "casado(a)", "Together": "morando junto", "Single": "solteiro(a)", "Divorced": "div
item_counts = df["estado_civil"].value_counts()
print(item_counts)
print(df)



# na coluna "estado_civil" foram encontradas 4 registros com as seguintes classificações:
# Absurd             2
# YOLO               2

# esses 4 registros são despreziveis


df.drop(df[df.estado_civil == 'Absurd' ].index, inplace=True)
df.drop(df[df.estado_civil == 'YOLO'   ].index, inplace=True)

item_counts = df["estado_civil"].value_counts()
print(item_counts)
print(df)


#df['id'].isna().sum()
for coluna in df.columns:
    print( coluna, '  Qtde de NaNs '   ,df[coluna].isna().sum())



# foram encontrados 24 rows com NaN de um total de 2241 rows do dataframe
# esses 24 rows representam  1,07% de toda base, esse valor está dentro da margem aceitavel para se dropada
```

```
df.dropna(inplace=True)
for coluna in df.columns:
    print( coluna, '  Qtde de NaNs '   ,df[coluna].isna().sum())
```

```
# instalando PySpark e algumas bibliotecas
!pip install pyspark
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType,StructField, StringType, IntegerType, FloatType, DoubleType, DateType
from pyspark.sql.functions import *
from pyspark     import SparkConf
import pyspark.sql.functions as F
```

```
# faz a conexão com a Sparksession

spark = (SparkSession.builder
                    .master("local")
                    .appName("atividade-individual")
                    .config('spark.ui.port', '4050')
                    .getOrCreate()
        )
```

```
# já que o df está tratado no pandas posso converter para um df em PySpark
spark.conf.set("spark.sql.execution.arrow.enabled","true")
dfspark = spark.createDataFrame(df)
```

```
# Realizar a mudança de nome de pelo menos 2 colunas

dfspark_2 = dfspark_2.withColumnRenamed("id","id_cliente").withColumnRenamed("ano_nascimento", "ano_de_nascimento")
dfspark_2.show(5)

# Deverá criar pelo menos duas novas colunas contendo alguma informação relevante sobre as outras colunas já existentes (F

# familias sem  filhos, sem criancas ou adolescentes

# criação da coluna total de filhos (qtde_criancas + qtde_adolescentes)
dfspark_2 = dfspark_2.withColumn('tot_de_filhos', F.col('qtde_criancas') + F.col('qtde_adolescentes'))

# criação da coluna número total de adesões de todas as campanhas
dfspark_2 = dfspark_2.withColumn('tot_adesao_camp_mkt', F.col('adesao_camp_mkt_1') +  F.col('adesao_camp_mkt_2') + F.col('


# usando windows function

from pyspark.sql.window import Window
from pyspark.sql.functions import row_number
windowSpec  = Window.partitionBy("educacao").orderBy("ano_de_nascimento")

#dfspark_2.withColumn("row_number",row_number().over(windowSpec)).show(truncate=False)
```