ProjetoCancelamentoClientes

November 29, 2024

[4]:
```
# Para manipulação e tratamento de dados
import numpy as np
import pandas as pd
import time
import matplotlib.pyplot as plt
import seaborn as sns

# Bibliotecas do Scikit Learn
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
from imblearn import under_sampling, over_sampling
from imblearn.over_sampling import SMOTE

# Para remover avisos de alerta
import warnings
warnings.filterwarnings('ignore')

# Para nao limitar a exibição do dataframe
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

0.1 Coleta de Dados

[6]:
```
df_original = pd.read_csv("dados.csv", sep = ';')
```

[9]:
```
df_original.shape
```

[9]:
```
(448447, 24)
```

[11]:
```
df_original.head()
```

[11]:
```
   ID_CLIENTE FORMA_AQUISICAO  IDADE_CLIENTE       SEXO  QT_FILHOS  \
0           1            Site             23  MASCULINO        0.0
```

```
1           2        Vendedor             24   FEMININO        0.0
2           3            Site             25  MASCULINO        0.0
3           4        Vendedor             26   FEMININO       17.0
4           5        Vendedor             27  MASCULINO        0.0

  DT_AQUISICAO DT_CANCELAMENTO  DIAS_ATIVO  MESES_ATIVO DURACAO_CONTRATO  \
0   18/06/2021             NaN          33            1         48 Meses
1   10/04/2018             NaN        1198           39         48 Meses
2   09/10/2020             NaN         285            9         48 Meses
3   25/06/2019             NaN         757           25         48 Meses
4   19/09/2019             NaN         671           22         48 Meses

   VL_PLANO_ADESAO  VL_PLANO_ATUAL                     NOME_PRODUTO  \
0              450             518     PLANO FAMILIA (100 CANAIS HD)
1              230             265       PLANO BASICO (30 CANAIS HD)
2              290             334  PLANO BASICO PLUS (50 CANAIS HD)
3              230             265       PLANO BASICO (30 CANAIS HD)
4              230             265       PLANO BASICO (30 CANAIS HD)

   QT_PONTOS_INSTALADOS  QT_PC_PAGAS  QT_PC_VENCIDAS  QT_PC_PAGA_ATRASO  \
0                     1            1               4                  0
1                     2            5               0                  1
2                     3            5               0                  3
3                     1            5               0                  1
4                     2            5               0                  5

   QT_PC_PAGA_EM_DIA  QT_ACORDO_PAGAMENTO  VL_MENSALIDADE_ATRASO  \
0                  1                    1                   2070
1                  4                    0                      0
2                  2                    0                      0
3                  4                    0                      0
4                  0                    0                      0

   VL_MENSALIDADE_DT_AQUISICAO  VL_MENSALIDADE_DT_ATUAL SITUACAO  COD_SITUACAO
0                          450                      518    ATIVO             0
1                          230                      265    ATIVO             0
2                          290                      334    ATIVO             0
3                          230                      265    ATIVO             0
4                          230                      265    ATIVO             0
```

[12]:
```
df_original.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 448447 entries, 0 to 448446
Data columns (total 24 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   ID_CLIENTE                   448447 non-null  int64
```

```
 1   FORMA_AQUISICAO              448447 non-null  object
 2   IDADE_CLIENTE                448447 non-null  int64
 3   SEXO                         448447 non-null  object
 4   QT_FILHOS                    448193 non-null  float64
 5   DT_AQUISICAO                 448447 non-null  object
 6   DT_CANCELAMENTO              117455 non-null  object
 7   DIAS_ATIVO                   448447 non-null  int64
 8   MESES_ATIVO                  448447 non-null  int64
 9   DURACAO_CONTRATO             448447 non-null  object
 10  VL_PLANO_ADESAO              448447 non-null  int64
 11  VL_PLANO_ATUAL               448447 non-null  int64
 12  NOME_PRODUTO                 448447 non-null  object
 13  QT_PONTOS_INSTALADOS         448447 non-null  int64
 14  QT_PC_PAGAS                  448447 non-null  int64
 15  QT_PC_VENCIDAS               448447 non-null  int64
 16  QT_PC_PAGA_ATRASO            448447 non-null  int64
 17  QT_PC_PAGA_EM_DIA            448447 non-null  int64
 18  QT_ACORDO_PAGAMENTO          448447 non-null  int64
 19  VL_MENSALIDADE_ATRASO        448447 non-null  int64
 20  VL_MENSALIDADE_DT_AQUISICAO  448447 non-null  int64
 21  VL_MENSALIDADE_DT_ATUAL      448447 non-null  int64
 22  SITUACAO                     448447 non-null  object
 23  COD_SITUACAO                 448447 non-null  int64
dtypes: float64(1), int64(16), object(7)
memory usage: 82.1+ MB
```

[13]:
```
# Avaliar o periodo dos dados coletados

inicio = pd.to_datetime(df_original['DT_AQUISICAO']).dt.date.min()
fim = pd.to_datetime(df_original['DT_AQUISICAO']).dt.date.max()
print(f'Periodo dos dados {inicio}, até {fim}')
```

```
Periodo dos dados 2001-01-01, até 2021-06-29
```

[14]:
```
df_original.describe()
```

[14]:

	ID_CLIENTE	IDADE_CLIENTE	QT_FILHOS	DIAS_ATIVO	\
count	448447.000000	448447.000000	448193.000000	448447.000000	
mean	224224.000000	38.891140	1.526385	483.857783	
std	129455.642421	6.682351	0.504288	373.649523	
min	1.000000	23.000000	0.000000	22.000000	
25%	112112.500000	35.000000	1.000000	167.000000	
50%	224224.000000	40.000000	2.000000	329.000000	
75%	336335.500000	43.000000	2.000000	798.000000	
max	448447.000000	55.000000	25.000000	1296.000000	

	MESES_ATIVO	VL_PLANO_ADESAO	VL_PLANO_ATUAL	QT_PONTOS_INSTALADOS	\
count	448447.000000	448447.000000	448447.000000	448447.000000	

```
mean       15.772457       303.769386      349.798596              1.625269
std        12.252344       113.612746      130.560943              0.579956
min         1.000000       230.000000      265.000000              1.000000
25%         5.000000       230.000000      265.000000              1.000000
50%        11.000000       230.000000      265.000000              2.000000
75%        26.000000       350.000000      403.000000              2.000000
max        42.000000       600.000000      690.000000              3.000000

         QT_PC_PAGAS  QT_PC_VENCIDAS  QT_PC_PAGA_ATRASO  QT_PC_PAGA_EM_DIA  \
count  448447.000000   448447.000000      448447.000000      448447.000000
mean        3.917281        1.019175           0.563596           3.353685
std         1.460079        1.474442           0.886314           1.586924
min         0.000000        0.000000           0.000000           0.000000
25%         3.000000        0.000000           0.000000           2.000000
50%         4.000000        0.000000           0.000000           4.000000
75%         5.000000        2.000000           1.000000           5.000000
max       100.000000       38.000000          17.000000         100.000000

       QT_ACORDO_PAGAMENTO  VL_MENSALIDADE_ATRASO  \
count        448447.000000          448447.000000
mean              0.188029             337.364498
std               0.423095             525.953927
min               0.000000               0.000000
25%               0.000000               0.000000
50%               0.000000               0.000000
75%               0.000000             794.000000
max               5.000000           16043.000000

       VL_MENSALIDADE_DT_AQUISICAO  VL_MENSALIDADE_DT_ATUAL   COD_SITUACAO
count                448447.000000           448447.000000  448447.000000
mean                   303.769386              349.798596       0.261915
std                    113.612746              130.560943       0.439677
min                    230.000000              265.000000       0.000000
25%                    230.000000              265.000000       0.000000
50%                    230.000000              265.000000       0.000000
75%                    350.000000              403.000000       1.000000
max                    600.000000              690.000000       1.000000
```

[15]:
```
# Verificando valores faltantes

print(df_original.isna().sum())
```

```
ID_CLIENTE                    0
FORMA_AQUISICAO               0
IDADE_CLIENTE                 0
SEXO                          0
QT_FILHOS                   254
DT_AQUISICAO                  0
```

```
DT_CANCELAMENTO                330992
DIAS_ATIVO                          0
MESES_ATIVO                         0
DURACAO_CONTRATO                    0
VL_PLANO_ADESAO                     0
VL_PLANO_ATUAL                      0
NOME_PRODUTO                        0
QT_PONTOS_INSTALADOS                0
QT_PC_PAGAS                         0
QT_PC_VENCIDAS                      0
QT_PC_PAGA_ATRASO                   0
QT_PC_PAGA_EM_DIA                   0
QT_ACORDO_PAGAMENTO                 0
VL_MENSALIDADE_ATRASO               0
VL_MENSALIDADE_DT_AQUISICAO         0
VL_MENSALIDADE_DT_ATUAL             0
SITUACAO                            0
COD_SITUACAO                        0
dtype: int64
```

[16]:
```
# Verificando valores unicos
df_original.nunique()
```

[16]:
```
ID_CLIENTE                    448447
FORMA_AQUISICAO                    2
IDADE_CLIENTE                     33
SEXO                               2
QT_FILHOS                          7
DT_AQUISICAO                    5888
DT_CANCELAMENTO                 5304
DIAS_ATIVO                      1051
MESES_ATIVO                       42
DURACAO_CONTRATO                   4
VL_PLANO_ADESAO                    6
VL_PLANO_ATUAL                     6
NOME_PRODUTO                       6
QT_PONTOS_INSTALADOS               3
QT_PC_PAGAS                       31
QT_PC_VENCIDAS                    31
QT_PC_PAGA_ATRASO                  9
QT_PC_PAGA_EM_DIA                 30
QT_ACORDO_PAGAMENTO                6
VL_MENSALIDADE_ATRASO             80
VL_MENSALIDADE_DT_AQUISICAO        6
VL_MENSALIDADE_DT_ATUAL            6
SITUACAO                           2
COD_SITUACAO                       2
```

```
dtype: int64
```

0.2 Análisando as Variáveis Categoricas(FORMA_AQUISICAO, SEXO, DURACAO_CONTRATO, NOME_PRODUTO, SITUACAO)

[17]:
```
# Agrupar os valores da variável
df_original.groupby(['FORMA_AQUISICAO']).size()
```

[17]:
```
FORMA_AQUISICAO
Site        321376
Vendedor    127071
dtype: int64
```

[18]:
```
df_original.groupby(['SEXO']).size()
```

[18]:
```
SEXO
FEMININO     224223
MASCULINO    224224
dtype: int64
```

[19]:
```
df_original.groupby(['DURACAO_CONTRATO']).size()
```

[19]:
```
DURACAO_CONTRATO
12 Meses       195
24 Meses       235
36 Meses     31889
48 Meses    416128
dtype: int64
```

[20]:
```
df_original.groupby(['NOME_PRODUTO']).size()
```

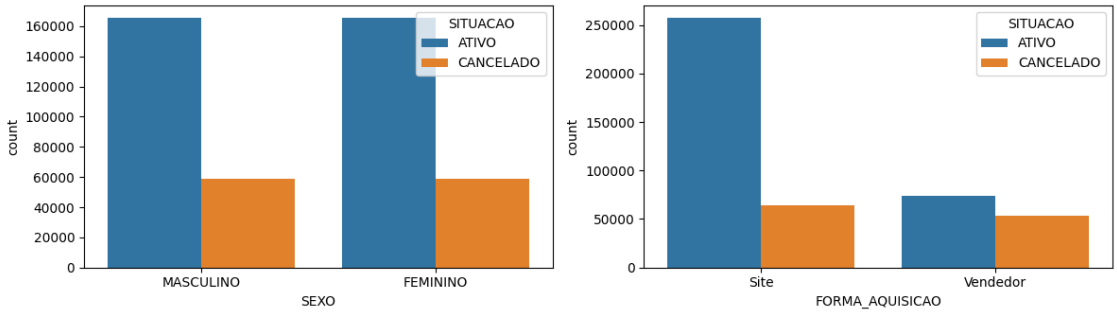
[20]:
```
NOME_PRODUTO
PLANO BASICO (30 CANAIS HD)          285209
PLANO BASICO PLUS (50 CANAIS HD)       8835
PLANO FAMILIA (100 CANAIS HD)         59716
PLANO MEDIO A (60 CANAIS HD)          62221
PLANO MEDIO TOP (90 CANAIS HD)          295
PLANO PREMIUM TOTAL                   32171
dtype: int64
```

[21]:
```
df_original.groupby(['SITUACAO']).size()
```

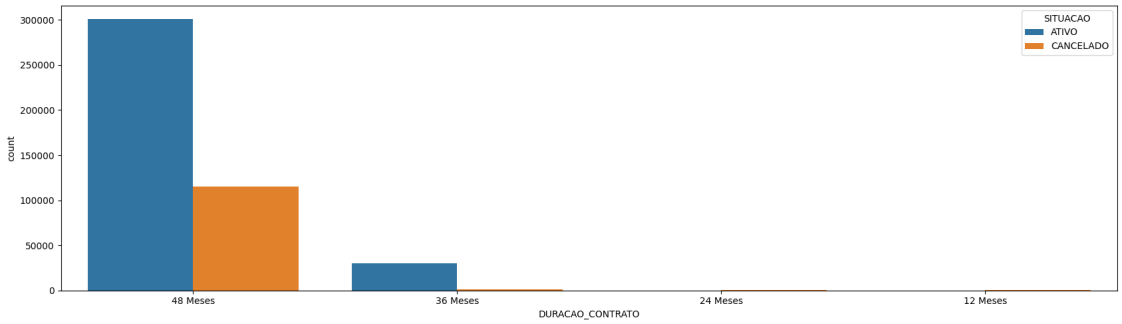
[21]:
```
SITUACAO
ATIVO        330992
CANCELADO    117455
dtype: int64
```

[23]:
```
# Analisando o grafico da variavel FORMA_AQUISICAO e SEXO comparadas a variavel alvo
# Podemos constatar na analise que não há discrepansias nestas variaveis

plt.rcParams['figure.figsize'] = [12.00, 3.50]
plt.rcParams['figure.autolayout'] = True
f, axes = plt.subplots(1,2)
sns.countplot(data = df_original, x = 'SEXO', hue='SITUACAO',  ax=axes[0])
sns.countplot(data = df_original, x = 'FORMA_AQUISICAO', hue='SITUACAO', ax=axes[1])
plt.show()
```

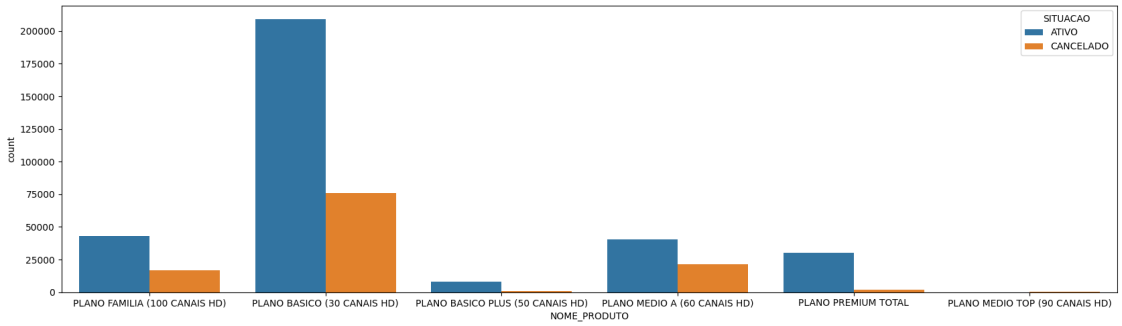


[24]:
```
# Analisando Gráfico da variavel DURACAO_CONTRATO comparadas coma variavel alvo
plt.rcParams['figure.figsize'] = [17.00, 5.00]
plt.rcParams['figure.autolayout'] = True
sns.countplot(data = df_original, x='DURACAO_CONTRATO', hue = 'SITUACAO')
plt.show()
```



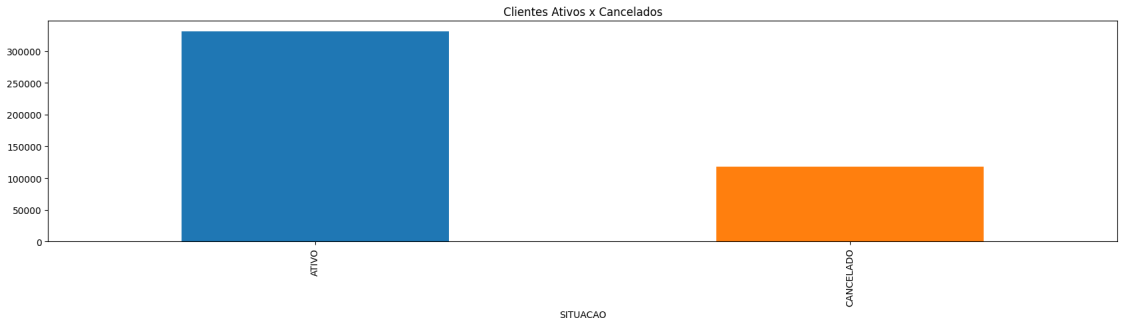
[25]:
```
# Analisando Gráfico da variavel NOME_PRODUTO comparadas coma variavel alvo
plt.rcParams['figure.figsize'] = [17.00, 5.00]
plt.rcParams['figure.autolayout'] = True
sns.countplot(data = df_original, x='NOME_PRODUTO', hue = 'SITUACAO')
```

```
plt.show()
```



[29]:
```
# Analisando como a variavel alvo esta distribuida
# Aqui podemos observar que há muito mais CLIENTES ATIVOS do que CLIENTES␣
 ↪CANCELADOS
# Dessa forma, precisaremos balancear o dataset
plt.rcParams['figure.figsize'] = [17.00, 5.00]
plt.rcParams['figure.autolayout'] = True
df_original.SITUACAO.value_counts().plot(kind='bar', title='Clientes Ativos x␣
 ↪Cancelados', color=['#1F77B4', '#FF7F0E'])
```

[29]:
```
<Axes: title={'center': 'Clientes Ativos x Cancelados'}, xlabel='SITUACAO'>
```



0.3 Analisando as Variáveis Numéricas

[32]:
```
# Carregar as variáveis para plot
variaveis_numericas = []
for i in df_original.columns[1:24].tolist():
    if df_original.dtypes[i] == 'int64' or df_original.dtypes[i] == 'float64':
        print(i, ':', df_original.dtypes[i])
        variaveis_numericas.append(i)
```

```
IDADE_CLIENTE : int64
QT_FILHOS : float64
DIAS_ATIVO : int64
MESES_ATIVO : int64
VL_PLANO_ADESAO : int64
VL_PLANO_ATUAL : int64
QT_PONTOS_INSTALADOS : int64
QT_PC_PAGAS : int64
QT_PC_VENCIDAS : int64
QT_PC_PAGA_ATRASO : int64
QT_PC_PAGA_EM_DIA : int64
QT_ACORDO_PAGAMENTO : int64
VL_MENSALIDADE_ATRASO : int64
VL_MENSALIDADE_DT_AQUISICAO : int64
VL_MENSALIDADE_DT_ATUAL : int64
COD_SITUACAO : int64
```

[33]:
```
# Quantidade de variaveis
len(variaveis_numericas)
```

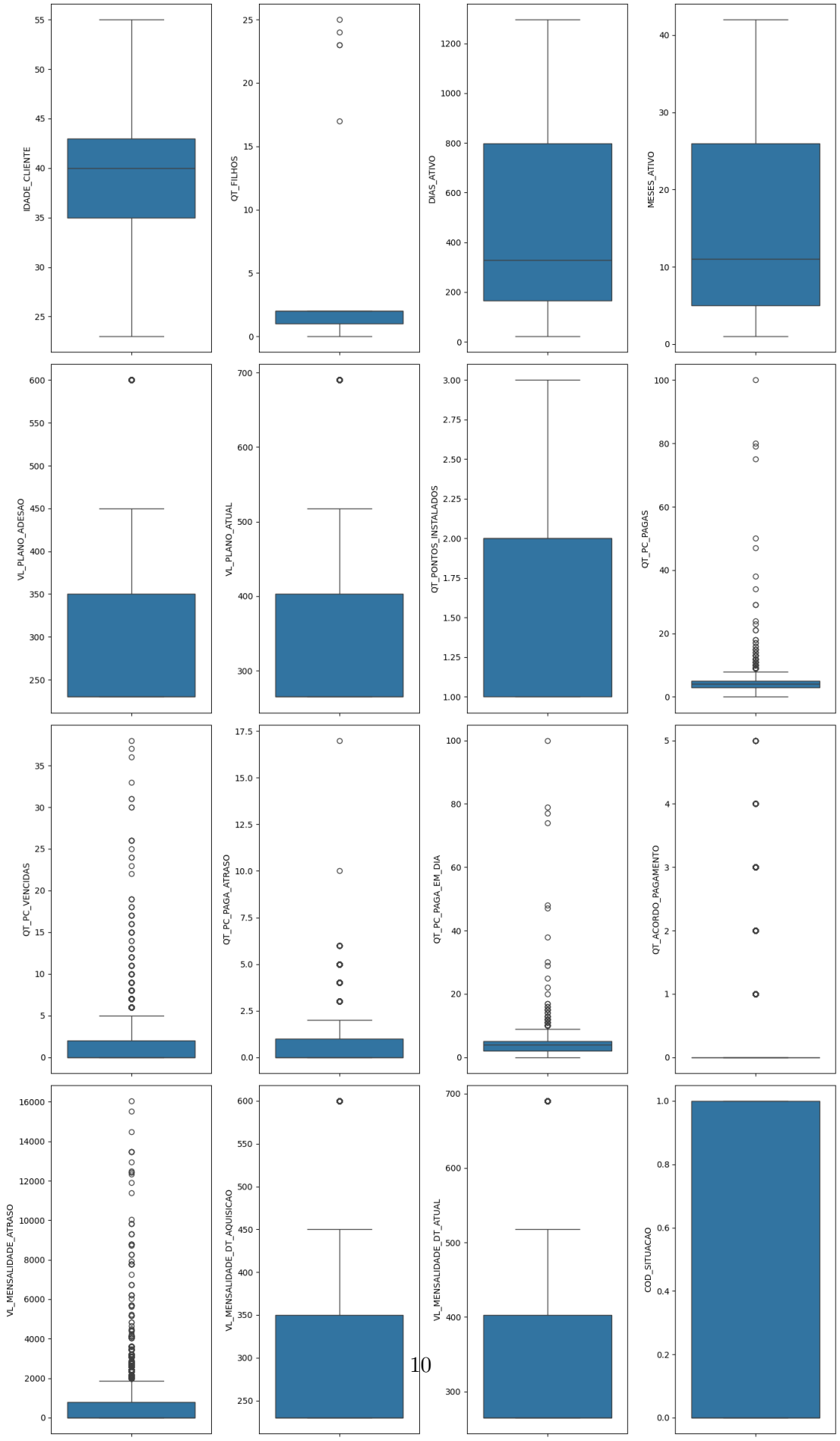
[33]: 16

[35]:
```
#Podemos observar nos boxplots abaixo que algumas variáveis númericas␣
 ↪apresentam uma grande quantidade de possiveis outliers
#Precisamos avaliar cada uma dessas variaveis dentro do contexto dos dados para␣
 ↪saber se realmente iremos tratala como outlier

plt.rcParams['figure.figsize'] = [14.00, 24.00]
plt.rcParams['figure.autolayout'] = True
f, axes = plt.subplots(4,4) # 4 linhas e 4 colunas

linha = 0
coluna = 0
for i in variaveis_numericas:
    sns.boxplot(data=df_original, y=i, ax=axes[linha][coluna])
    coluna+=1
    if coluna == 4:
        linha += 1
        coluna= 0

plt.show()
```


IDADE_CLIENTE
QT_FILHOS
DIAS_ATIVO
MESES_ATIVO
VL_PLANO_ADESAO
VL_PLANO_ATUAL
QT_PONTOS_INSTALADOS
QT_PC_PAGAS
QT_PC_VENCIDAS
QT_PC_PAGA_ATRASO
QT_PC_PAGA_EM_DIA
QT_ACORDO_PAGAMENTO
VL_MENSALIDADE_ATRASO
VL_MENSALIDADE_DT_AQUISICAO
VL_MENSALIDADE_DT_ATUAL
COD_SITUACAO

[36]:
```
# A variavel QT_PC_PAGA_EM_DIA e QT_PC_PAGAS possui um numero maior que o prazo␣
 ↪maximo de contrato
#por isso podemos constatar que foi algum erro dos dados gerando este OUTLIER e␣
 ↪iremos trata-los considerando o numero máximo do prazo do contrato

# A Variavel QT_FILHOS também possui alguns OUTLIERS como por exemplo, 17, 23 e␣
 ↪24 filhos
# vamos avaliar a quantidade desses dados e verificar como iremos trata-los
```

[39]:
```
# Temos apenas 5 registros dentro de um volume de mais de 440 mil registros,␣
 ↪dessa forma iremos exlui-los
# pois nao irá causar impacto de perda de dados, pois a quantidade é muito␣
 ↪pequena comparada ao todo
df_original.groupby(df_original['QT_FILHOS']).size()
```

[39]:
```
QT_FILHOS
0.0         10
1.0     212353
2.0     235825
17.0         1
23.0         2
24.0         1
25.0         1
dtype: int64
```

[40]:
```
# Aqui podemos ver os registros..
df_original.loc[df_original['QT_FILHOS'] > 2]
```

[40]:
```
      ID_CLIENTE FORMA_AQUISICAO  IDADE_CLIENTE       SEXO  QT_FILHOS  \
3              4        Vendedor             26   FEMININO       17.0
91            92        Vendedor             48   FEMININO       23.0
164          165        Vendedor             55  MASCULINO       23.0
273          274        Vendedor             32   FEMININO       24.0
454          455        Vendedor             38  MASCULINO       25.0

    DT_AQUISICAO DT_CANCELAMENTO  DIAS_ATIVO  MESES_ATIVO DURACAO_CONTRATO  \
3     25/06/2019             NaN         757           25         48 Meses
91    03/08/2018             NaN        1083           35         48 Meses
164   19/06/2018             NaN        1128           37         48 Meses
273   02/05/2018             NaN        1176           38         48 Meses
454   04/09/2018             NaN        1051           34         48 Meses

     VL_PLANO_ADESAO  VL_PLANO_ATUAL                 NOME_PRODUTO  \
3                230             265  PLANO BASICO (30 CANAIS HD)
```

```
91                230              265  PLANO BASICO (30 CANAIS HD)
164               230              265  PLANO BASICO (30 CANAIS HD)
273               230              265  PLANO BASICO (30 CANAIS HD)
454               230              265  PLANO BASICO (30 CANAIS HD)

     QT_PONTOS_INSTALADOS  QT_PC_PAGAS  QT_PC_VENCIDAS  QT_PC_PAGA_ATRASO  \
3                       1            5               0                  1
91                      2            5               0                  1
164                     3            5               0                  2
273                     1            5               0                  2
454                     2            5               0                  1

     QT_PC_PAGA_EM_DIA  QT_ACORDO_PAGAMENTO  VL_MENSALIDADE_ATRASO  \
3                    4                    0                      0
91                   4                    0                      0
164                  3                    1                      0
273                  3                    1                      0
454                  4                    1                      0

     VL_MENSALIDADE_DT_AQUISICAO  VL_MENSALIDADE_DT_ATUAL SITUACAO  \
3                            230                      265    ATIVO
91                           230                      265    ATIVO
164                          230                      265    ATIVO
273                          230                      265    ATIVO
454                          230                      265    ATIVO

     COD_SITUACAO
3               0
91              0
164             0
273             0
454             0
```

1 Tratamento de Dados

Vamos tratar os dados que identificamos na fase de Analise Exploratoria

- Tratar OUTLIERS
- Tratar valores Nulos(Váriavel QT_FILHOS possui 254 valores nulos)
- Tratar as variaveis QT_PC_PAGA_EM_DIA e QT_PC_PAGAS
- Engenharia de Atributos (Criar variavel NIVEL PAGAMENTO)
- Balancear a Variável target
- Aplicar o OneHotEncoding

[41]:
```
# Manteremos o DataFrame original e os dados tratados ficarão no DataFrame␣
  ↪chamado df_dados
df_dados = df_original.loc[df_original['QT_FILHOS'] <= 2]
```

```
df_dados.shape
```

[41]:
```
(448188, 24)
```

[42]:
```
df_dados.groupby(['QT_FILHOS']).size()
```

[42]:
```
QT_FILHOS
0.0        10
1.0    212353
2.0    235825
dtype: int64
```

[43]:
```
print('Media de filhos:', df_dados['QT_FILHOS'].mean())
print('Mediana de filhos:', df_dados['QT_FILHOS'].median())
print('Moda de filhos:', df_dados['QT_FILHOS'].mode())
```

```
Media de filhos: 1.5261519719403465
Mediana de filhos: 2.0
Moda de filhos: 0    2.0
Name: QT_FILHOS, dtype: float64
Moda de filhos: 0    2.0
Name: QT_FILHOS, dtype: float64
```

[44]:
```
# Preencheremos os valores NULOS com a mediana dos dados
df_dados['QT_FILHOS'] = df_dados['QT_FILHOS'].fillna((df_dados["QT_FILHOS"].
  ↪median()))
df_dados.isnull().sum()
```

[44]:
```
ID_CLIENTE                     0
FORMA_AQUISICAO                0
IDADE_CLIENTE                  0
SEXO                           0
QT_FILHOS                      0
DT_AQUISICAO                   0
DT_CANCELAMENTO           330987
DIAS_ATIVO                     0
MESES_ATIVO                    0
DURACAO_CONTRATO               0
VL_PLANO_ADESAO                0
VL_PLANO_ATUAL                 0
NOME_PRODUTO                   0
QT_PONTOS_INSTALADOS           0
QT_PC_PAGAS                    0
QT_PC_VENCIDAS                 0
QT_PC_PAGA_ATRASO              0
QT_PC_PAGA_EM_DIA              0
QT_ACORDO_PAGAMENTO            0
VL_MENSALIDADE_ATRASO          0
```

```
VL_MENSALIDADE_DT_AQUISICAO         0
VL_MENSALIDADE_DT_ATUAL             0
SITUACAO                            0
COD_SITUACAO                        0
dtype: int64
```

[45]:
```
# Substituindo os dados da variavel DURACAO_CONTRATO para mantermos somente os␣
 ↪números
df_dados['DURACAO_CONTRATO'] = df_dados['DURACAO_CONTRATO'].replace(['12␣
 ↪Meses'], 12)
df_dados['DURACAO_CONTRATO'] = df_dados['DURACAO_CONTRATO'].replace(['24␣
 ↪Meses'], 24)
df_dados['DURACAO_CONTRATO'] = df_dados['DURACAO_CONTRATO'].replace(['36␣
 ↪Meses'], 36)
df_dados['DURACAO_CONTRATO'] = df_dados['DURACAO_CONTRATO'].replace(['48␣
 ↪Meses'], 48)
```

[46]:
```
#Visualizando as primeiras linhas do datase
df_dados.head()
```

[46]:

	ID_CLIENTE	FORMA_AQUISICAO	IDADE_CLIENTE	SEXO	QT_FILHOS
0	1	Site	23	MASCULINO	0.0
1	2	Vendedor	24	FEMININO	0.0
2	3	Site	25	MASCULINO	0.0
4	5	Vendedor	27	MASCULINO	0.0
5	6	Vendedor	28	FEMININO	1.0

	DT_AQUISICAO	DT_CANCELAMENTO	DIAS_ATIVO	MESES_ATIVO	DURACAO_CONTRATO
0	18/06/2021	NaN	33	1	48
1	10/04/2018	NaN	1198	39	48
2	09/10/2020	NaN	285	9	48
4	19/09/2019	NaN	671	22	48
5	23/03/2018	NaN	1216	40	48

	VL_PLANO_ADESAO	VL_PLANO_ATUAL	NOME_PRODUTO
0	450	518	PLANO FAMILIA (100 CANAIS HD)
1	230	265	PLANO BASICO (30 CANAIS HD)
2	290	334	PLANO BASICO PLUS (50 CANAIS HD)
4	230	265	PLANO BASICO (30 CANAIS HD)
5	230	265	PLANO BASICO (30 CANAIS HD)

	QT_PONTOS_INSTALADOS	QT_PC_PAGAS	QT_PC_VENCIDAS	QT_PC_PAGA_ATRASO
0	1	1	4	0
1	2	5	0	1
2	3	5	0	3
4	2	5	0	5
5	3	5	0	0

```
   QT_PC_PAGA_EM_DIA  QT_ACORDO_PAGAMENTO  VL_MENSALIDADE_ATRASO  \
0                  1                    1                   2070
1                  4                    0                      0
2                  2                    0                      0
4                  0                    0                      0
5                  5                    0                      0

   VL_MENSALIDADE_DT_AQUISICAO  VL_MENSALIDADE_DT_ATUAL SITUACAO  COD_SITUACAO
0                          450                      518    ATIVO             0
1                          230                      265    ATIVO             0
2                          290                      334    ATIVO             0
4                          230                      265    ATIVO             0
5                          230                      265    ATIVO             0
```

[47]:
```
#Visualizando as informações dos tipos de variáveis
df_dados.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 448188 entries, 0 to 448446
Data columns (total 24 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   ID_CLIENTE                   448188 non-null  int64
 1   FORMA_AQUISICAO              448188 non-null  object
 2   IDADE_CLIENTE                448188 non-null  int64
 3   SEXO                         448188 non-null  object
 4   QT_FILHOS                    448188 non-null  float64
 5   DT_AQUISICAO                 448188 non-null  object
 6   DT_CANCELAMENTO              117201 non-null  object
 7   DIAS_ATIVO                   448188 non-null  int64
 8   MESES_ATIVO                  448188 non-null  int64
 9   DURACAO_CONTRATO             448188 non-null  int64
 10  VL_PLANO_ADESAO              448188 non-null  int64
 11  VL_PLANO_ATUAL               448188 non-null  int64
 12  NOME_PRODUTO                 448188 non-null  object
 13  QT_PONTOS_INSTALADOS         448188 non-null  int64
 14  QT_PC_PAGAS                  448188 non-null  int64
 15  QT_PC_VENCIDAS               448188 non-null  int64
 16  QT_PC_PAGA_ATRASO            448188 non-null  int64
 17  QT_PC_PAGA_EM_DIA            448188 non-null  int64
 18  QT_ACORDO_PAGAMENTO          448188 non-null  int64
 19  VL_MENSALIDADE_ATRASO        448188 non-null  int64
 20  VL_MENSALIDADE_DT_AQUISICAO  448188 non-null  int64
 21  VL_MENSALIDADE_DT_ATUAL      448188 non-null  int64
 22  SITUACAO                     448188 non-null  object
 23  COD_SITUACAO                 448188 non-null  int64
dtypes: float64(1), int64(17), object(6)
```

```
memory usage: 85.5+ MB
```

[48]:
```
# Identificando as quantidades máximas para tratar os OUTLIERS
print(df_dados['QT_PC_PAGAS'].max())
print(df_dados['QT_PC_PAGA_EM_DIA'].max())
```

```
100
100
```

[51]:
```
# Registros que possuirem as quantidades superiores a duração do contrato,␣
 ↪iremos alterar e colocar o numero de duração do contrato
df_dados.loc[df_dados.QT_PC_PAGAS > df_dados.DURACAO_CONTRATO, 'QT_PC_PAGAS'] =␣
 ↪df_dados['DURACAO_CONTRATO']
df_dados.loc[df_dados.QT_PC_PAGA_EM_DIA > df_dados.DURACAO_CONTRATO,␣
 ↪'QT_PC_PAGA_EM_DIA'] = df_dados['DURACAO_CONTRATO']
```

[52]:
```
#Verificando se as variaveis foram ajustadas
print(df_dados['QT_PC_PAGAS'].max())
print(df_dados['QT_PC_PAGA_EM_DIA'].max())
```

```
48
48
```

[53]:
```
# Engenharia de atributos
# Criando uma nova varipávelde categoria de nivel de pagamento de acordo com a␣
 ↪quantidade de parcelas pagas
bins = [-100, 3, 6, 12, 48]
labels = ['RUIM', 'MEDIO', 'BOM', 'OTIMO']
df_dados['NIVEL_PAGAMENTO'] = pd.cut(df_dados['QT_PC_PAGAS'], bins=bins,␣
 ↪labels=labels)
pd.value_counts(df_dados.NIVEL_PAGAMENTO)
```

[53]:
```
NIVEL_PAGAMENTO
MEDIO    297750
RUIM     149912
BOM         488
OTIMO        38
Name: count, dtype: int64
```

[54]:
```
df_dados.head()
```

[54]:

	ID_CLIENTE	FORMA_AQUISICAO	IDADE_CLIENTE	SEXO	QT_FILHOS	\
0	1	Site	23	MASCULINO	0.0	
1	2	Vendedor	24	FEMININO	0.0	
2	3	Site	25	MASCULINO	0.0	
4	5	Vendedor	27	MASCULINO	0.0	
5	6	Vendedor	28	FEMININO	1.0	

```
  DT_AQUISICAO DT_CANCELAMENTO  DIAS_ATIVO  MESES_ATIVO  DURACAO_CONTRATO  \
0   18/06/2021             NaN          33            1                48
1   10/04/2018             NaN        1198           39                48
2   09/10/2020             NaN         285            9                48
4   19/09/2019             NaN         671           22                48
5   23/03/2018             NaN        1216           40                48

   VL_PLANO_ADESAO  VL_PLANO_ATUAL                    NOME_PRODUTO  \
0              450             518     PLANO FAMILIA (100 CANAIS HD)
1              230             265       PLANO BASICO (30 CANAIS HD)
2              290             334  PLANO BASICO PLUS (50 CANAIS HD)
4              230             265       PLANO BASICO (30 CANAIS HD)
5              230             265       PLANO BASICO (30 CANAIS HD)

   QT_PONTOS_INSTALADOS  QT_PC_PAGAS  QT_PC_VENCIDAS  QT_PC_PAGA_ATRASO  \
0                     1            1               4                  0
1                     2            5               0                  1
2                     3            5               0                  3
4                     2            5               0                  5
5                     3            5               0                  0

   QT_PC_PAGA_EM_DIA  QT_ACORDO_PAGAMENTO  VL_MENSALIDADE_ATRASO  \
0                  1                    1                   2070
1                  4                    0                      0
2                  2                    0                      0
4                  0                    0                      0
5                  5                    0                      0

   VL_MENSALIDADE_DT_AQUISICAO  VL_MENSALIDADE_DT_ATUAL SITUACAO  \
0                          450                      518    ATIVO
1                          230                      265    ATIVO
2                          290                      334    ATIVO
4                          230                      265    ATIVO
5                          230                      265    ATIVO

   COD_SITUACAO NIVEL_PAGAMENTO
0             0            RUIM
1             0           MEDIO
2             0           MEDIO
4             0           MEDIO
5             0           MEDIO
```

[55]:
```
#Fazendo uma cópia do DataFrame
df_dados_2 = df_dados.copy()
df_dados_2.head()
```

[55]:

	ID_CLIENTE	FORMA_AQUISICAO	IDADE_CLIENTE	SEXO	QT_FILHOS
0	1	Site	23	MASCULINO	0.0
1	2	Vendedor	24	FEMININO	0.0
2	3	Site	25	MASCULINO	0.0
4	5	Vendedor	27	MASCULINO	0.0
5	6	Vendedor	28	FEMININO	1.0

	DT_AQUISICAO	DT_CANCELAMENTO	DIAS_ATIVO	MESES_ATIVO	DURACAO_CONTRATO
0	18/06/2021	NaN	33	1	48
1	10/04/2018	NaN	1198	39	48
2	09/10/2020	NaN	285	9	48
4	19/09/2019	NaN	671	22	48
5	23/03/2018	NaN	1216	40	48

	VL_PLANO_ADESAO	VL_PLANO_ATUAL	NOME_PRODUTO
0	450	518	PLANO FAMILIA (100 CANAIS HD)
1	230	265	PLANO BASICO (30 CANAIS HD)
2	290	334	PLANO BASICO PLUS (50 CANAIS HD)
4	230	265	PLANO BASICO (30 CANAIS HD)
5	230	265	PLANO BASICO (30 CANAIS HD)

	QT_PONTOS_INSTALADOS	QT_PC_PAGAS	QT_PC_VENCIDAS	QT_PC_PAGA_ATRASO
0	1	1	4	0
1	2	5	0	1
2	3	5	0	3
4	2	5	0	5
5	3	5	0	0

	QT_PC_PAGA_EM_DIA	QT_ACORDO_PAGAMENTO	VL_MENSALIDADE_ATRASO
0	1	1	2070
1	4	0	0
2	2	0	0
4	0	0	0
5	5	0	0

	VL_MENSALIDADE_DT_AQUISICAO	VL_MENSALIDADE_DT_ATUAL	SITUACAO
0	450	518	ATIVO
1	230	265	ATIVO
2	290	334	ATIVO
4	230	265	ATIVO
5	230	265	ATIVO

	COD_SITUACAO	NIVEL_PAGAMENTO
0	0	RUIM
1	0	MEDIO
2	0	MEDIO
4	0	MEDIO

```
5              0           MEDIO
```

[56]:
```
# Cria o encoder
lb = LabelEncoder()

# Aplica o encoder nas variáveis que estão com string
# O encoder irá transformar essas variáveis emnúmeros(lembre-se, os algoritmos␣
 ↪de Machine Learning so entendem numeros)
df_dados_2['SEXO'] = lb.fit_transform(df_dados_2['SEXO'])
df_dados_2['FORMA_AQUISICAO'] = lb.fit_transform(df_dados_2['FORMA_AQUISICAO'])
df_dados_2['NOME_PRODUTO'] = lb.fit_transform(df_dados_2['NOME_PRODUTO'])
df_dados_2['NIVEL_PAGAMENTO'] = lb.fit_transform(df_dados_2['NIVEL_PAGAMENTO'])
```

[57]:
```
df_dados_2.head(20)
```

[57]:

	ID_CLIENTE	FORMA_AQUISICAO	IDADE_CLIENTE	SEXO	QT_FILHOS	DT_AQUISICAO	\
0	1	0	23	1	0.0	18/06/2021	
1	2	1	24	0	0.0	10/04/2018	
2	3	0	25	1	0.0	09/10/2020	
4	5	1	27	1	0.0	19/09/2019	
5	6	1	28	0	1.0	23/03/2018	
6	7	1	29	1	2.0	04/02/2019	
7	8	0	30	0	1.0	11/11/2020	
8	9	1	31	1	2.0	31/07/2018	
9	10	0	32	0	1.0	03/05/2021	
10	11	0	33	1	2.0	16/06/2021	
11	12	0	34	0	1.0	26/03/2021	
12	13	0	35	1	2.0	20/02/2021	
13	14	0	36	0	1.0	18/12/2020	
14	15	0	37	1	2.0	18/12/2020	
15	16	0	38	0	1.0	13/11/2020	
16	17	1	39	1	2.0	07/02/2019	
17	18	1	40	0	1.0	04/02/2019	
18	19	0	41	1	2.0	17/05/2021	
19	20	1	42	0	1.0	17/04/2020	
20	21	0	43	1	2.0	05/11/2020	

	DT_CANCELAMENTO	DIAS_ATIVO	MESES_ATIVO	DURACAO_CONTRATO	\
0	NaN	33	1	48	
1	NaN	1198	39	48	
2	NaN	285	9	48	
4	NaN	671	22	48	
5	NaN	1216	40	48	
6	NaN	898	29	48	
7	NaN	252	8	48	
8	NaN	1086	36	48	
9	NaN	79	2	48	

10	NaN	35	1	48
11	NaN	117	4	48
12	NaN	151	5	48
13	NaN	215	7	48
14	NaN	215	7	48
15	NaN	250	8	48
16	NaN	895	29	48
17	NaN	898	29	48
18	NaN	65	2	48
19	NaN	460	15	48
20	NaN	258	8	48

	VL_PLANO_ADESAO	VL_PLANO_ATUAL	NOME_PRODUTO	QT_PONTOS_INSTALADOS	\
0	450	518	2	1	
1	230	265	0	2	
2	290	334	1	3	
4	230	265	0	2	
5	230	265	0	3	
6	350	403	3	1	
7	230	265	0	2	
8	290	334	1	3	
9	230	265	0	1	
10	230	265	0	2	
11	230	265	0	3	
12	230	265	0	1	
13	230	265	0	2	
14	230	265	0	3	
15	230	265	0	1	
16	230	265	0	2	
17	350	403	3	3	
18	230	265	0	1	
19	230	265	0	2	
20	230	265	0	3	

	QT_PC_PAGAS	QT_PC_VENCIDAS	QT_PC_PAGA_ATRASO	QT_PC_PAGA_EM_DIA	\
0	1	4	0	1	
1	5	0	1	4	
2	5	0	3	2	
4	5	0	5	0	
5	5	0	0	5	
6	1	0	0	1	
7	5	0	2	3	
8	4	0	2	2	
9	2	3	1	1	
10	1	5	0	1	
11	4	1	0	4	
12	4	2	4	0	

13	3	0	2	1
14	3	0	2	1
15	6	0	0	6
16	4	0	3	1
17	1	0	0	1
18	2	4	1	1
19	3	0	1	2
20	2	0	0	2

	QT_ACORDO_PAGAMENTO	VL_MENSALIDADE_ATRASO	VL_MENSALIDADE_DT_AQUISICAO	\
0	1	2070	450	
1	0	0	230	
2	0	0	290	
4	0	0	230	
5	0	0	230	
6	0	0	350	
7	1	0	230	
8	0	0	290	
9	1	794	230	
10	1	1323	230	
11	1	265	230	
12	0	529	230	
13	0	0	230	
14	0	0	230	
15	1	0	230	
16	1	0	230	
17	0	0	350	
18	0	1058	230	
19	0	0	230	
20	1	0	230	

	VL_MENSALIDADE_DT_ATUAL	SITUACAO	COD_SITUACAO	NIVEL_PAGAMENTO
0	518	ATIVO	0	3
1	265	ATIVO	0	1
2	334	ATIVO	0	1
4	265	ATIVO	0	1
5	265	ATIVO	0	1
6	403	ATIVO	0	3
7	265	ATIVO	0	1
8	334	ATIVO	0	1
9	265	ATIVO	0	3
10	265	ATIVO	0	3
11	265	ATIVO	0	1
12	265	ATIVO	0	1
13	265	ATIVO	0	3
14	265	ATIVO	0	3
15	265	ATIVO	0	1

```
16                    265    ATIVO           0               1
17                    403    ATIVO           0               3
18                    265    ATIVO           0               3
19                    265    ATIVO           0               3
20                    265    ATIVO           0               3
```

[58]:
```
# Listando as colunas do nosso DataSet
df_dados_2.columns.tolist()
```

[58]:
```
['ID_CLIENTE',
 'FORMA_AQUISICAO',
 'IDADE_CLIENTE',
 'SEXO',
 'QT_FILHOS',
 'DT_AQUISICAO',
 'DT_CANCELAMENTO',
 'DIAS_ATIVO',
 'MESES_ATIVO',
 'DURACAO_CONTRATO',
 'VL_PLANO_ADESAO',
 'VL_PLANO_ATUAL',
 'NOME_PRODUTO',
 'QT_PONTOS_INSTALADOS',
 'QT_PC_PAGAS',
 'QT_PC_VENCIDAS',
 'QT_PC_PAGA_ATRASO',
 'QT_PC_PAGA_EM_DIA',
 'QT_ACORDO_PAGAMENTO',
 'VL_MENSALIDADE_ATRASO',
 'VL_MENSALIDADE_DT_AQUISICAO',
 'VL_MENSALIDADE_DT_ATUAL',
 'SITUACAO',
 'COD_SITUACAO',
 'NIVEL_PAGAMENTO']
```

[59]:
```
# Vamos filtrar e utilizar somente as colunas necessárias
columns = ['FORMA_AQUISICAO',
 'IDADE_CLIENTE',
 'SEXO',
 'QT_FILHOS',


 'DIAS_ATIVO',
 'MESES_ATIVO',
 'DURACAO_CONTRATO',
 'VL_PLANO_ADESAO',
 'VL_PLANO_ATUAL',
```

```
 'NOME_PRODUTO',
 'QT_PONTOS_INSTALADOS',
 'QT_PC_PAGAS',
 'QT_PC_VENCIDAS',
 'QT_PC_PAGA_ATRASO',
 'QT_PC_PAGA_EM_DIA',
 'QT_ACORDO_PAGAMENTO',
 'VL_MENSALIDADE_ATRASO',
 'VL_MENSALIDADE_DT_AQUISICAO',
 'VL_MENSALIDADE_DT_ATUAL',

 'NIVEL_PAGAMENTO',
 'COD_SITUACAO']

df_dados_2 = pd.DataFrame(df_dados_2, columns=columns)
```

[60]:
```
df_dados_2.head()
```

[60]:

	FORMA_AQUISICAO	IDADE_CLIENTE	SEXO	QT_FILHOS	DIAS_ATIVO	MESES_ATIVO	\
0	0	23	1	0.0	33	1	
1	1	24	0	0.0	1198	39	
2	0	25	1	0.0	285	9	
4	1	27	1	0.0	671	22	
5	1	28	0	1.0	1216	40	

	DURACAO_CONTRATO	VL_PLANO_ADESAO	VL_PLANO_ATUAL	NOME_PRODUTO	\
0	48	450	518	2	
1	48	230	265	0	
2	48	290	334	1	
4	48	230	265	0	
5	48	230	265	0	

	QT_PONTOS_INSTALADOS	QT_PC_PAGAS	QT_PC_VENCIDAS	QT_PC_PAGA_ATRASO	\
0	1	1	4	0	
1	2	5	0	1	
2	3	5	0	3	
4	2	5	0	5	
5	3	5	0	0	

	QT_PC_PAGA_EM_DIA	QT_ACORDO_PAGAMENTO	VL_MENSALIDADE_ATRASO	\
0	1	1	2070	
1	4	0	0	
2	2	0	0	
4	0	0	0	
5	5	0	0	

	VL_MENSALIDADE_DT_AQUISICAO	VL_MENSALIDADE_DT_ATUAL	NIVEL_PAGAMENTO	\

```
0                          450                        518                3
1                          230                        265                1
2                          290                        334                1
4                          230                        265                1
5                          230                        265                1

   COD_SITUACAO
0             0
1             0
2             0
4             0
5             0
```

[61]:
```
#Verificando os tipos de variaveis
df_dados_2.info()
```

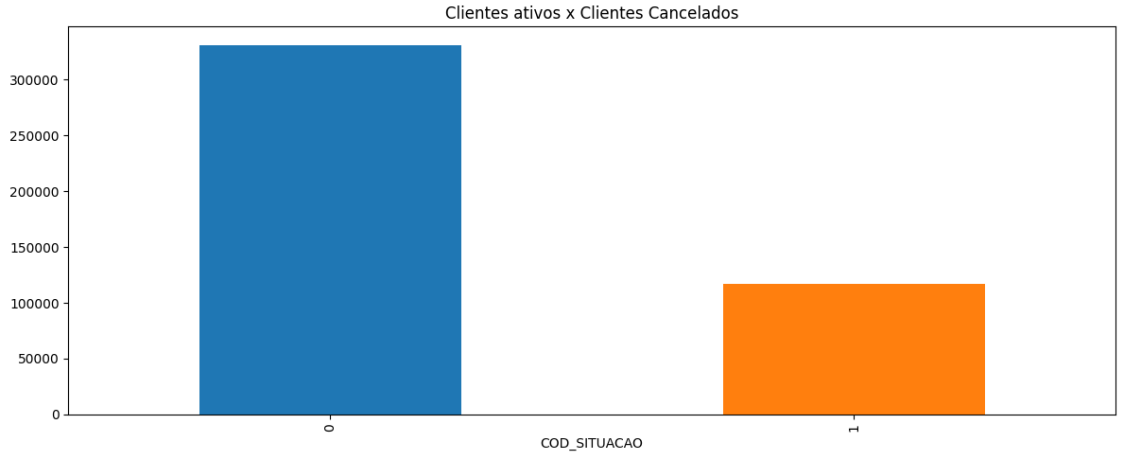
```
<class 'pandas.core.frame.DataFrame'>
Index: 448188 entries, 0 to 448446
Data columns (total 21 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   FORMA_AQUISICAO               448188 non-null  int64
 1   IDADE_CLIENTE                 448188 non-null  int64
 2   SEXO                          448188 non-null  int64
 3   QT_FILHOS                     448188 non-null  float64
 4   DIAS_ATIVO                    448188 non-null  int64
 5   MESES_ATIVO                   448188 non-null  int64
 6   DURACAO_CONTRATO              448188 non-null  int64
 7   VL_PLANO_ADESAO               448188 non-null  int64
 8   VL_PLANO_ATUAL                448188 non-null  int64
 9   NOME_PRODUTO                  448188 non-null  int64
 10  QT_PONTOS_INSTALADOS          448188 non-null  int64
 11  QT_PC_PAGAS                   448188 non-null  int64
 12  QT_PC_VENCIDAS                448188 non-null  int64
 13  QT_PC_PAGA_ATRASO             448188 non-null  int64
 14  QT_PC_PAGA_EM_DIA             448188 non-null  int64
 15  QT_ACORDO_PAGAMENTO           448188 non-null  int64
 16  VL_MENSALIDADE_ATRASO         448188 non-null  int64
 17  VL_MENSALIDADE_DT_AQUISICAO   448188 non-null  int64
 18  VL_MENSALIDADE_DT_ATUAL       448188 non-null  int64
 19  NIVEL_PAGAMENTO               448188 non-null  int64
 20  COD_SITUACAO                  448188 non-null  int64
dtypes: float64(1), int64(20)
memory usage: 91.4 MB
```

[65]:
```
# Avaliando como a variavel alvo esta distribuida
# Aqui podemos observar que há muito mais CLIENTES ATIVOS do que CLIENTES␣
 ↪CANCELADO
```

```
# dessa forma, precisaremos balancear o dataset na etapa 2 - Tratamento de Dados

plt.rcParams['figure.figsize'] = [12.00,5.00]
plt.rcParams['figure.autolayout'] = True
df_dados_2.COD_SITUACAO.value_counts().plot(kind='bar', title='Clientes ativos
 ↪x Clientes Cancelados', color=['#1F77B4', '#FF7F0E'])
```

[65]:
```
<Axes: title={'center': 'Clientes ativos x Clientes Cancelados'},
xlabel='COD_SITUACAO'>
```



[66]:
```
# Separar variaveis preditoras e target
PREDITORAS = df_dados_2.iloc[:, 0:20]
TARGET = df_dados_2.iloc[:, 20]
```

[67]:
```
PREDITORAS.head()
```

[67]:

	FORMA_AQUISICAO	IDADE_CLIENTE	SEXO	QT_FILHOS	DIAS_ATIVO	MESES_ATIVO	\
0	0	23	1	0.0	33	1	
1	1	24	0	0.0	1198	39	
2	0	25	1	0.0	285	9	
4	1	27	1	0.0	671	22	
5	1	28	0	1.0	1216	40	

	DURACAO_CONTRATO	VL_PLANO_ADESAO	VL_PLANO_ATUAL	NOME_PRODUTO	\
0	48	450	518	2	
1	48	230	265	0	
2	48	290	334	1	
4	48	230	265	0	
5	48	230	265	0	

	QT_PONTOS_INSTALADOS	QT_PC_PAGAS	QT_PC_VENCIDAS	QT_PC_PAGA_ATRASO	\

```
0                     1             1                4                   0
1                     2             5                0                   1
2                     3             5                0                   3
4                     2             5                0                   5
5                     3             5                0                   0

   QT_PC_PAGA_EM_DIA  QT_ACORDO_PAGAMENTO  VL_MENSALIDADE_ATRASO  \
0                  1                    1                   2070
1                  4                    0                      0
2                  2                    0                      0
4                  0                    0                      0
5                  5                    0                      0

   VL_MENSALIDADE_DT_AQUISICAO  VL_MENSALIDADE_DT_ATUAL  NIVEL_PAGAMENTO
0                          450                      518                3
1                          230                      265                1
2                          290                      334                1
4                          230                      265                1
5                          230                      265                1
```

[68]:
```
TARGET.head()
```

[68]:
```
0    0
1    0
2    0
4    0
5    0
Name: COD_SITUACAO, dtype: int64
```

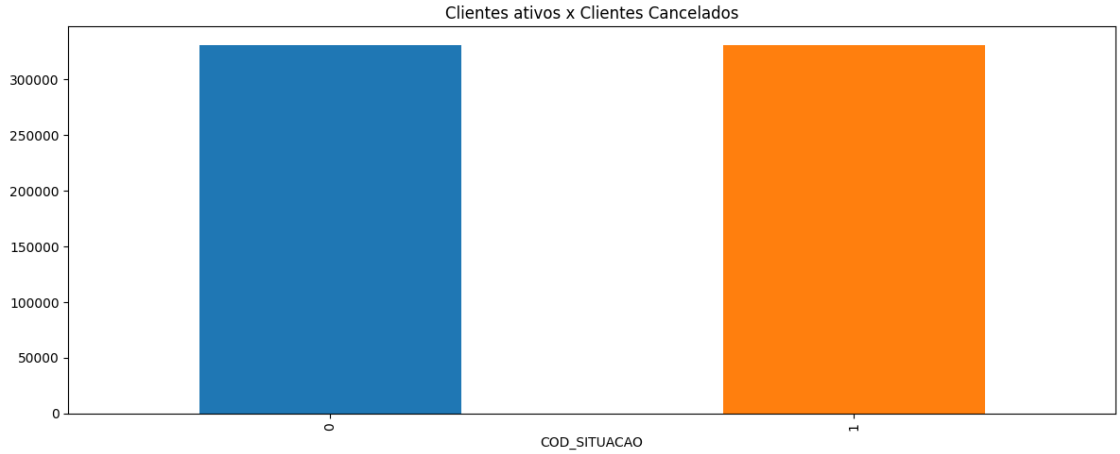
[69]:
```
#Seed para reproduzir o mesmo resultado
seed = 100

#Cria o balanceador SMOTE
balanceador = SMOTE(random_state=seed)

#Aplica o Balanceador
PREDITORAS_RES, TARGET_RES =  balanceador.fit_resample(PREDITORAS, TARGET)
```

[72]:
```
# Visualizando o balanceamento da variável TARGET
plt.rcParams['figure.figsize'] = [12.00,5.00]
plt.rcParams['figure.autolayout'] = True
TARGET_RES.value_counts().plot(kind='bar', title='Clientes ativos x Clientes␣
 ↪Cancelados', color=['#1F77B4', '#FF7F0E'])
```

[72]:
```
<Axes: title={'center': 'Clientes ativos x Clientes Cancelados'},
xlabel='COD_SITUACAO'>
```



[73]:
```
# Quantidade  de registros antes do balanceamento
PREDITORAS.shape
```

[73]: (448188, 20)

[74]:
```
# Quantidade de registros antes do balanceamento
TARGET.shape
```

[74]: (448188,)

[75]:
```
# Quantidade de registros apos o balanceamento
PREDITORAS_RES.shape
```

[75]: (661974, 20)

[76]:
```
# Quantidade de registros apos o balanceamento
TARGET_RES.shape
```

[76]: (661974,)

1.1 Criação e Avaliação do modelo preditivo

[82]:
```
# Divisão em dados de Treino e teste
X_treino, X_teste, Y_treino, Y_teste = train_test_split(PREDITORAS_RES,␣
 ↪TARGET_RES, test_size=0.3, random_state=42)
```

[83]:
```
# Padronizando as Variáveis - Pré Processamento de dados
Padronizador = StandardScaler()
X_treino_padronizados = Padronizador.fit_transform(X_treino)
X_teste_padronizados = Padronizador.transform(X_teste)
```

[84]:
```
# visualizando os dados padronizados
X_treino_padronizados
```

[84]:
```
array([[-0.69344562,  0.53255991,  1.06022683, …,  1.41114896,
         1.41223978, -0.94605613],
       [ 1.44207415, -0.02658459, -0.94319439, …,  1.41114896,
         1.41223978,  1.05546166],
       [-0.69344562, -0.02658459, -0.94319439, …,  2.81634037,
         2.81422044, -0.94605613],
       …,
       [-0.69344562, -1.56423196, -0.94319439, …, -0.64979845,
        -0.64997596, -0.94605613],
       [ 1.44207415,  0.53255991, -0.94319439, …,  0.47435468,
         0.47486899,  1.05546166],
       [ 1.44207415,  0.53255991, -0.94319439, …,  1.41114896,
         1.41223978, -0.94605613]])
```

[85]:
```
# Range de valores de k que iremos testar
kVals = range(3,10,2)
```

[86]:
```
# Lista vazia para receber acurácias
acuracias = []
```

[87]:
```
start = time.time()
for k in kVals:

    # Treinando o modelo KNN com cada valor de k
    modeloKNN = KNeighborsClassifier(n_neighbors=k)
    modeloKNN.fit(X_treino_padronizados, Y_treino)

    # Avaliando o modelo e atualizando a Lista de Acurácias
    score = modeloKNN.score(X_teste_padronizados, Y_teste)
    print("Com valor de k = %d, a acurácia é = %.2f%%" % (k, score * 100))
    acuracias.append(score)
end = time.time()
print('Tempo de treinamento do modelo:', end - start)
```

```
Com valor de k = 3, a acurácia é = 97.69%
Com valor de k = 5, a acurácia é = 97.38%
Com valor de k = 7, a acurácia é = 97.20%
Com valor de k = 9, a acurácia é = 97.03%
Tempo de treinamento do modelo: 480.9965007305145
```

[88]:
```
# Obtendo o valor de k que apresentou a maior acurácia
i = np.argmax(acuracias)
print('O valor de k = %d alcançou a mais alta acurácia de %.2f%% nos dados de␣
 ↪validação' % (kVals[i],acuracias[i] * 100))
```

```
O valor de k = 3 alcançou a mais alta acurácia de 97.69% nos dados de validação
```

[89]:
```
# Criando a versão final do modelo com o maior valor de k
modeloFinal = KNeighborsClassifier(n_neighbors=kVals[i])
```

[90]:
```
#Treinamento do modelo
#Observe que neste caso usou a metrica de distancia de minkowski mas isso␣
 ↪podemos mudar os parametros
modeloFinal.fit(X_treino_padronizados, Y_treino)
```

[90]:
```
KNeighborsClassifier(n_neighbors=3)
```

[91]:
```
# Previsões com os dados de teste
previsoes = modeloFinal.predict(X_teste_padronizados)
```

[92]:
```
print('Acuracia do modelo: ', accuracy_score(Y_teste, previsoes))
```

```
Acuracia do modelo:  0.9769327216971394
```

[ ]:
```

```