2º Projeto Prático - Dados - Análise de Dados e Predição com Python e Panda

João Vitor de Oliveira Ribas

jvoribas@gmail.com

2024

Introdução

Como parte do programa Desenvolve Boticário 2024, a seguir é apresentada a análise de uma base de dados de preços de aluguéis em São Paulo.

A análise compreendeu todos as etapas necessárias, como importação dos dados, limpeza, exploração, visualização e conclusões a partir dos dados.

Também foi realizado o ajuste de um modelo preditivo por meio de regressão linear, assim como a avaliação de seu desempenho.

Metodologia

A análise dos dados foi realizada pelas seguintes etapas:

1. Importação das Bibliotecas
2. Importação dos Dados
3. Análise Exploratória dos Dados
4. Tratamento dos Dados
5. Preparação dos Dados
6. Ajuste dos Modelos (Regressão Linear, Regressão Linear(Huber), Random Forest e Gradient Boosting Tree)
7. Conclusão

0. Importação das Bibliotecas

```
import numpy as np
import pandas as pd
import seaborn as sns
import xgboost as xgb
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LinearRegression, LogisticRegression,
SGDRegressor, HuberRegressor
from sklearn.metrics import mean_squared_error, r2_score
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor,
GradientBoostingRegressor
```

1. Importação dos Dados

```
df_bruto = pd.read_csv('base-alugueis-sp.csv')
```

2. Análise Exploratória dos Dados

```
df_bruto.head()
```

```
              address      district  area  bedrooms  garage  \
0          Rua Herval    Belenzinho    21         1       0
1  Avenida São Miguel  Vila Marieta    15         1       1
2    Rua Oscar Freire     Pinheiros    18         1       0
3    Rua Júlio Sayago       Vila Ré    56         2       2
4  Rua Barata Ribeiro    Bela Vista    19         1       0

                  type  rent  total
0     Studio e kitnet  2400   2939
1     Studio e kitnet  1030   1345
2         Apartamento  4000   4661
3  Casa em condomínio  1750   1954
4     Studio e kitnet  4000   4654
```

```
df_bruto.shape
```

```
(11657, 8)
```

```
df_bruto.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11657 entries, 0 to 11656
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   address   11657 non-null  object
 1   district  11657 non-null  object
 2   area      11657 non-null  int64
 3   bedrooms  11657 non-null  int64
 4   garage    11657 non-null  int64
 5   type      11657 non-null  object
 6   rent      11657 non-null  int64
```

```
 7   total     11657 non-null  int64
dtypes: int64(5), object(3)
memory usage: 728.7+ KB
```

```
df_bruto.describe().round(2)
```

```
           area  bedrooms    garage      rent     total
count  11657.00  11657.00  11657.00  11657.00  11657.00
mean      84.66      1.97      1.06   3250.81   4080.03
std       74.02      0.93      1.13   2650.71   3352.48
min        0.00      0.00      0.00    500.00    509.00
25%       40.00      1.00      0.00   1590.00   1996.00
50%       60.00      2.00      1.00   2415.00   3057.00
75%       96.00      3.00      2.00   3800.00   4774.00
max      580.00      6.00      6.00  25000.00  28700.00
```

```
df_bruto.isnull().sum()
```

```
address     0
district    0
area        0
bedrooms    0
garage      0
type        0
rent        0
total       0
dtype: int64
```

```
df_bruto.isna().sum()
```

```
address     0
district    0
area        0
bedrooms    0
garage      0
type        0
rent        0
total       0
dtype: int64
```

```
df_bruto = df_bruto.rename(columns={'address': 'Endereço', 'district':
'Distrito', 'area': 'Area', 'bedrooms': 'Quartos', 'garage':
'Garagem', 'type': 'Tipo', 'rent': 'Aluguel', 'total': 'Total'})
```

```
df_bruto.head()
```

```
             Endereço      Distrito  Area  Quartos  Garagem  \
0          Rua Herval    Belenzinho    21        1        0
1  Avenida São Miguel  Vila Marieta    15        1        1
2    Rua Oscar Freire     Pinheiros    18        1        0
3    Rua Júlio Sayago       Vila Ré    56        2        2
```

```
4  Rua Barata Ribeiro    Bela Vista    19        1        0
```

```
                  Tipo  Aluguel  Total
0     Studio e kitnet     2400   2939
1     Studio e kitnet     1030   1345
2         Apartamento     4000   4661
3  Casa em condomínio     1750   1954
4     Studio e kitnet     4000   4654
```

Como o conjunto de dados não possui documentação, então não sabemos exatamente o que a coluna 'Total' representa.

Por esse motivo, vamos considerar que caso o imóvel possua taxa de condomínio, esse valor já estará agregado ao valor do aluguel.

Dessa forma, vamos definir que a coluna 'Total' representa o valor do aluguel com a adição do valor do IPTU.

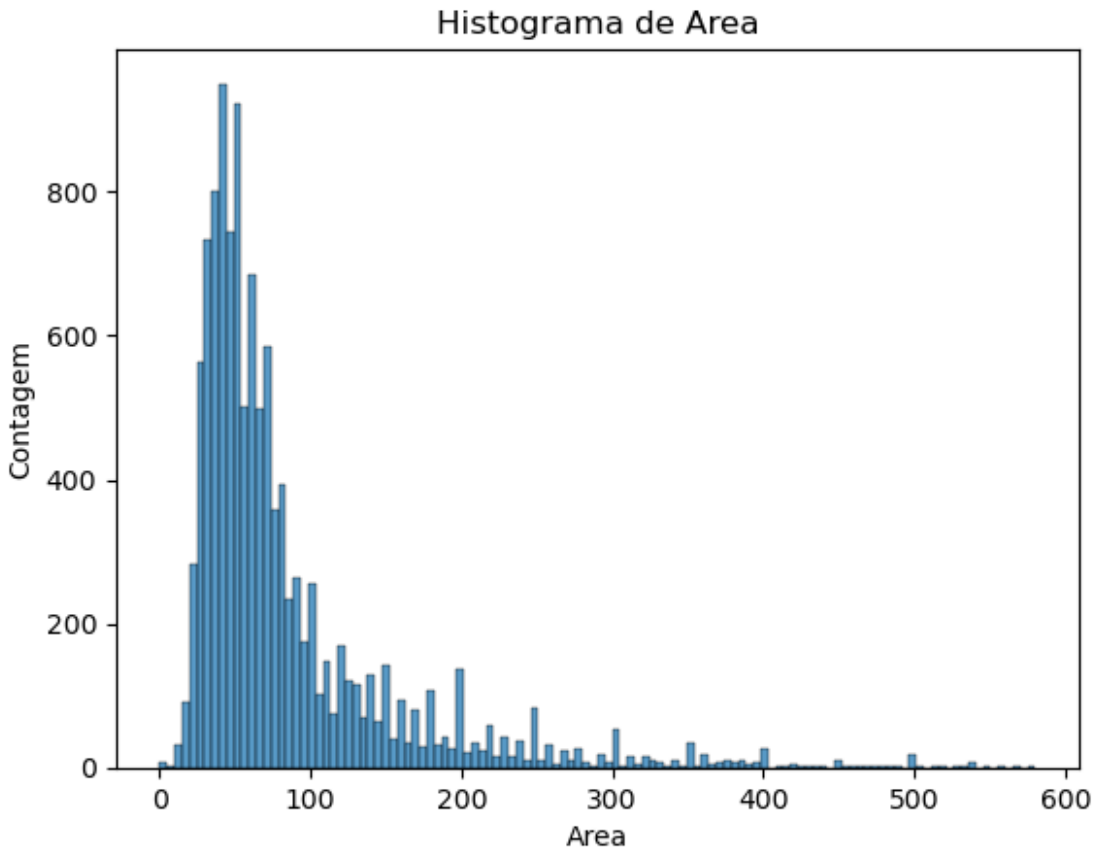
Então vamos criar essa nova categoria chamada 'IPTU' e remover a 'Total', pois as categorias restantes são mais úteis.

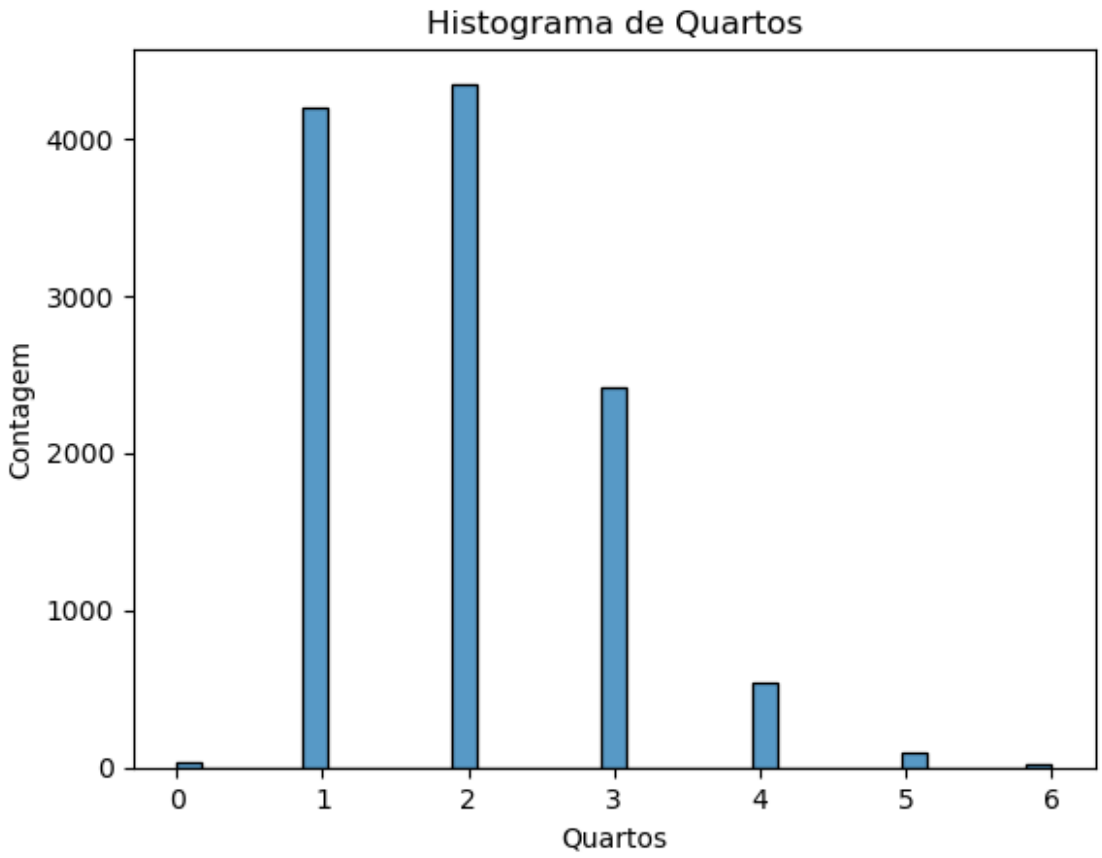
```
df_bruto['IPTU'] = df_bruto['Total'] - df_bruto['Aluguel']

df_bruto.drop('Total', axis=1, inplace=True)

df_bruto.head()
```

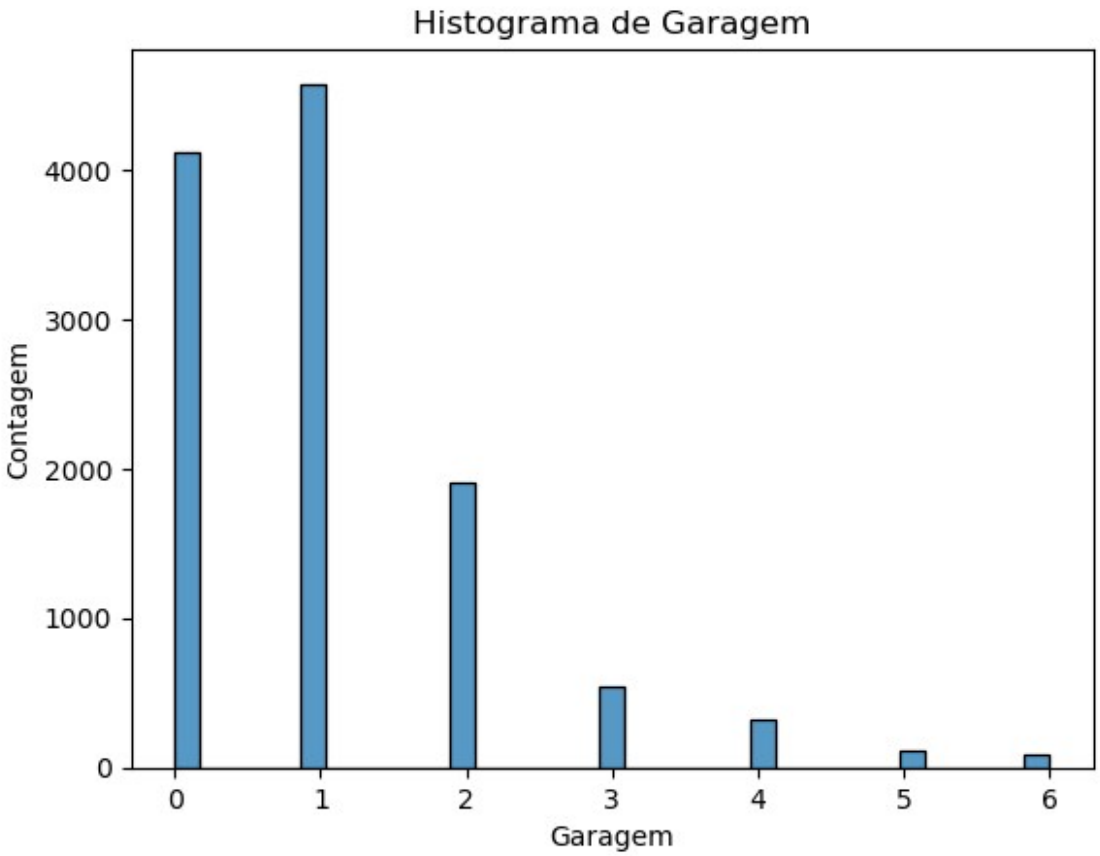
```
             Endereço      Distrito  Area  Quartos  Garagem  \
0          Rua Herval    Belenzinho    21        1        0
1  Avenida São Miguel  Vila Marieta    15        1        1
2    Rua Oscar Freire     Pinheiros    18        1        0
3    Rua Júlio Sayago       Vila Ré    56        2        2
4  Rua Barata Ribeiro    Bela Vista    19        1        0
```

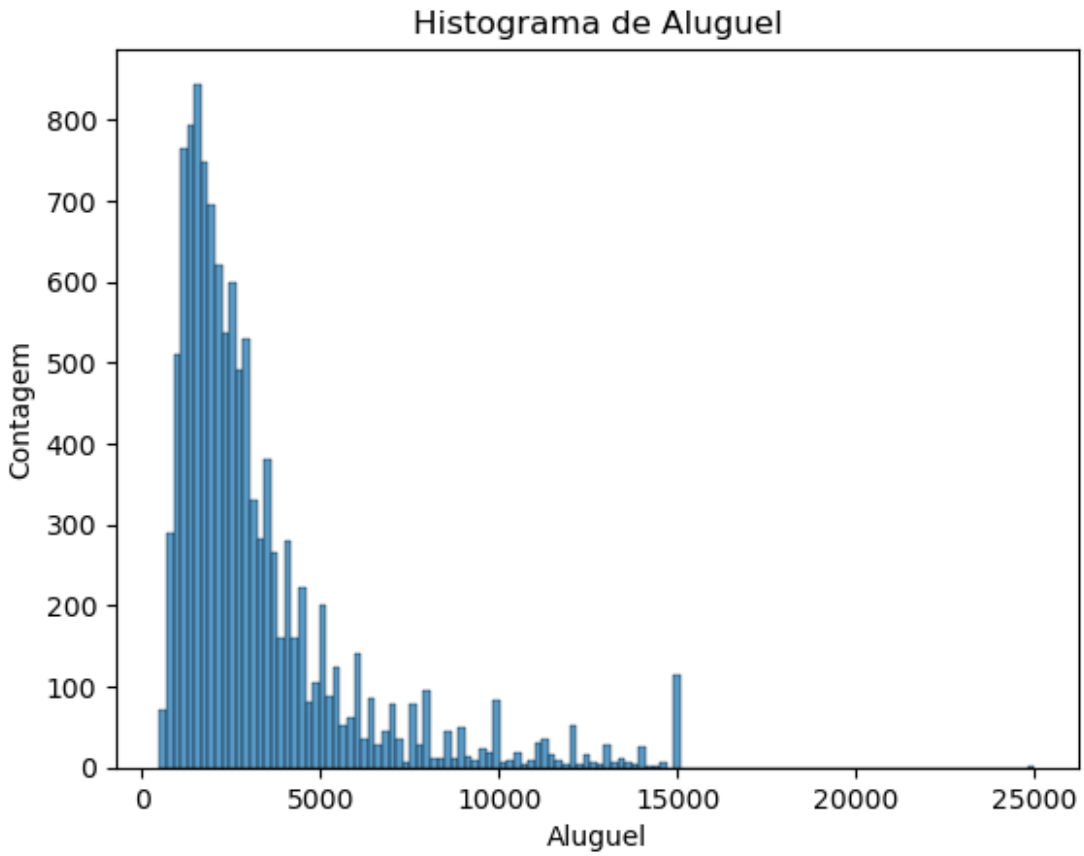
```
                  Tipo  Aluguel  IPTU
0     Studio e kitnet     2400   539
1     Studio e kitnet     1030   315
2         Apartamento     4000   661
3  Casa em condomínio     1750   204
4     Studio e kitnet     4000   654
```

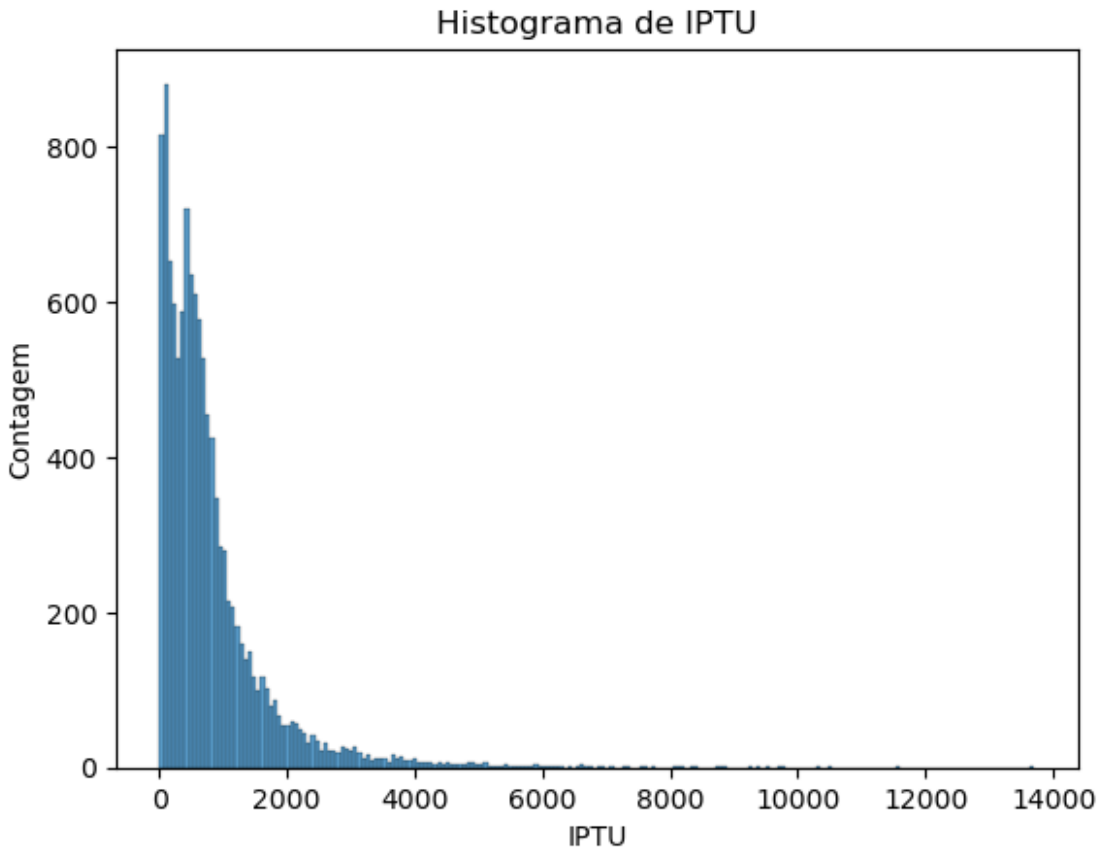
```
colunas_quantitativas = ['Area', 'Quartos', 'Garagem', 'Aluguel',
'IPTU']

for column in colunas_quantitativas:
    sns.histplot(df_bruto[column])
    plt.title(f'Histograma de {column}')
    plt.ylabel('Contagem')
    plt.show()
```


Histograma de Area
Contagem
0
200
400
600
800
0
100
200
300
400
500
600
Area


Histograma de Quartos
Contagem
Quartos
0
1000
2000
3000
4000
0
1
2
3
4
5
6


Histograma de Garagem
Contagem
0
1000
2000
3000
4000
0
1
2
3
4
5
6
Garagem


Histograma de Aluguel
Contagem
Aluguel
0
100
200
300
400
500
600
700
800
0
5000
10000
15000
20000
25000



Pelos histogramas gerais, não obtivemos nenhuma grande informação a respeito do conjunto de dados.

Então vamos separar o conjunto pelos tipos de imóveis e observar o restultado.

```
df_studio_kitnet = df_bruto.query('Tipo == "Studio e kitnet"')
print(df_studio_kitnet.shape)
df_studio_kitnet.head()
```

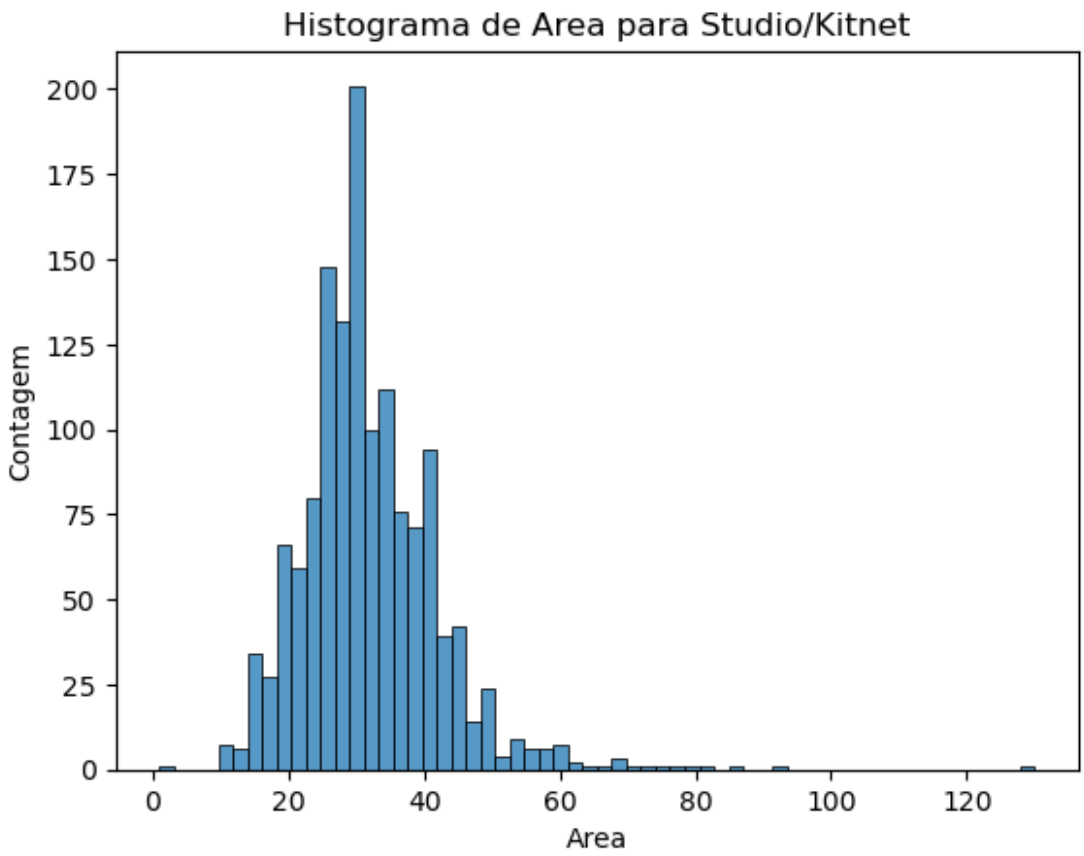
```
(1381, 8)
```

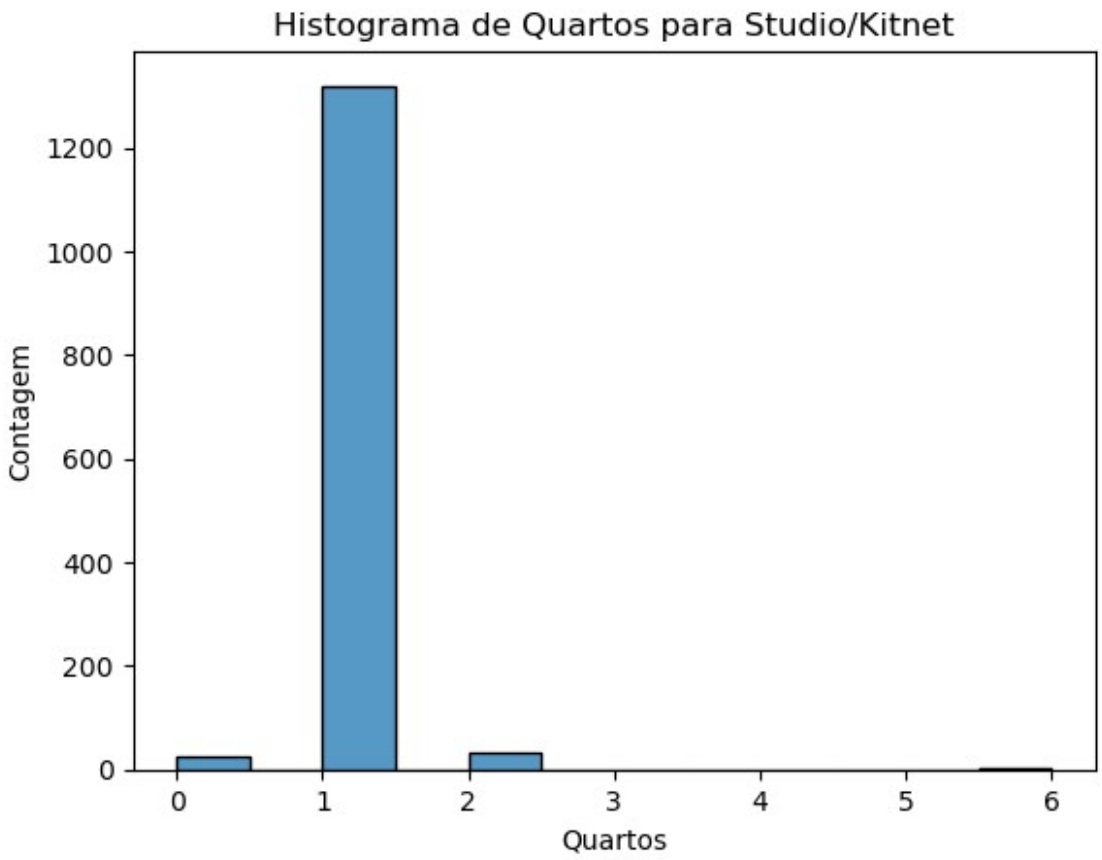
	Endereço	Distrito	Area	Quartos	Garagem	Tipo	Aluguel	IPTU
0	Rua Herval	Belenzinho	21	1	0	Studio e kitnet	2400	539
1	Avenida São Miguel	Vila Marieta	15	1	1	Studio e kitnet	1030	315
4	Rua Barata Ribeiro	Bela Vista	19	1	0	Studio e kitnet	4000	654
7	Avenida Cásper Líbero	Centro	26	1	0	Studio e kitnet	1727	517
12	Rua Henrique Sertório	Tatuapé	32	1	0	Studio e kitnet	2100	498

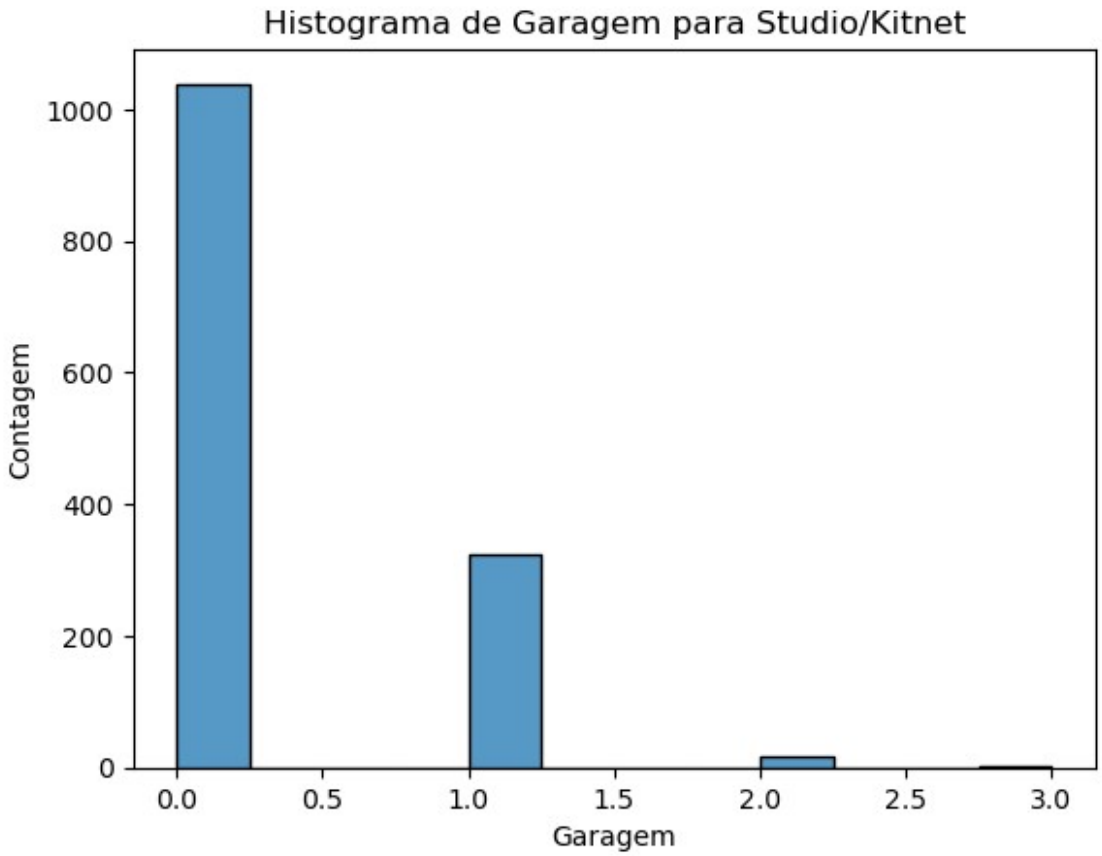
```
df_studio_kitnet.describe()
```

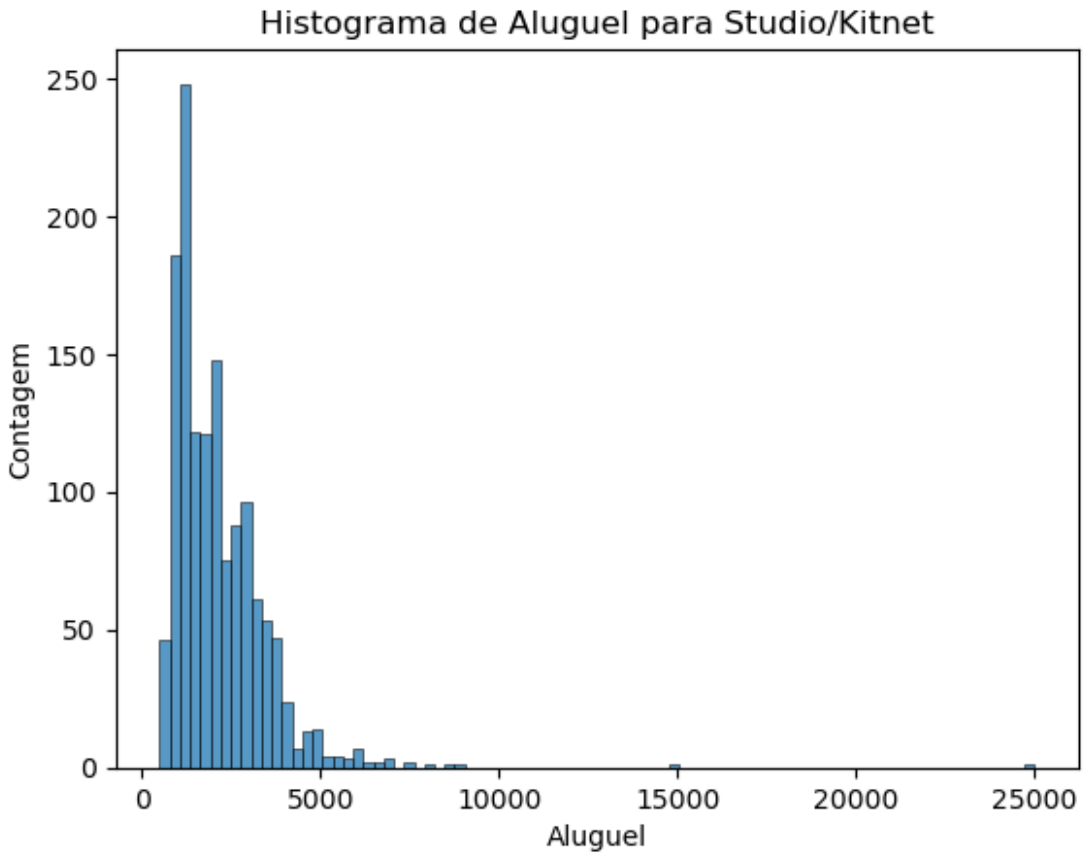
```
              Area      Quartos      Garagem       Aluguel
IPTU
count  1381.000000  1381.000000  1381.000000   1381.000000
1381.000000
mean     31.742216     1.009413     0.260681   2127.825489
540.454743
std      10.125382     0.248003     0.469470   1365.744349
403.077090
min       1.000000     0.000000     0.000000    500.000000
13.000000
25%      25.000000     1.000000     0.000000   1200.000000
274.000000
50%      30.000000     1.000000     0.000000   1850.000000
489.000000
75%      37.000000     1.000000     0.000000   2790.000000
702.000000
max     130.000000     6.000000     3.000000  25000.000000
3391.000000
```

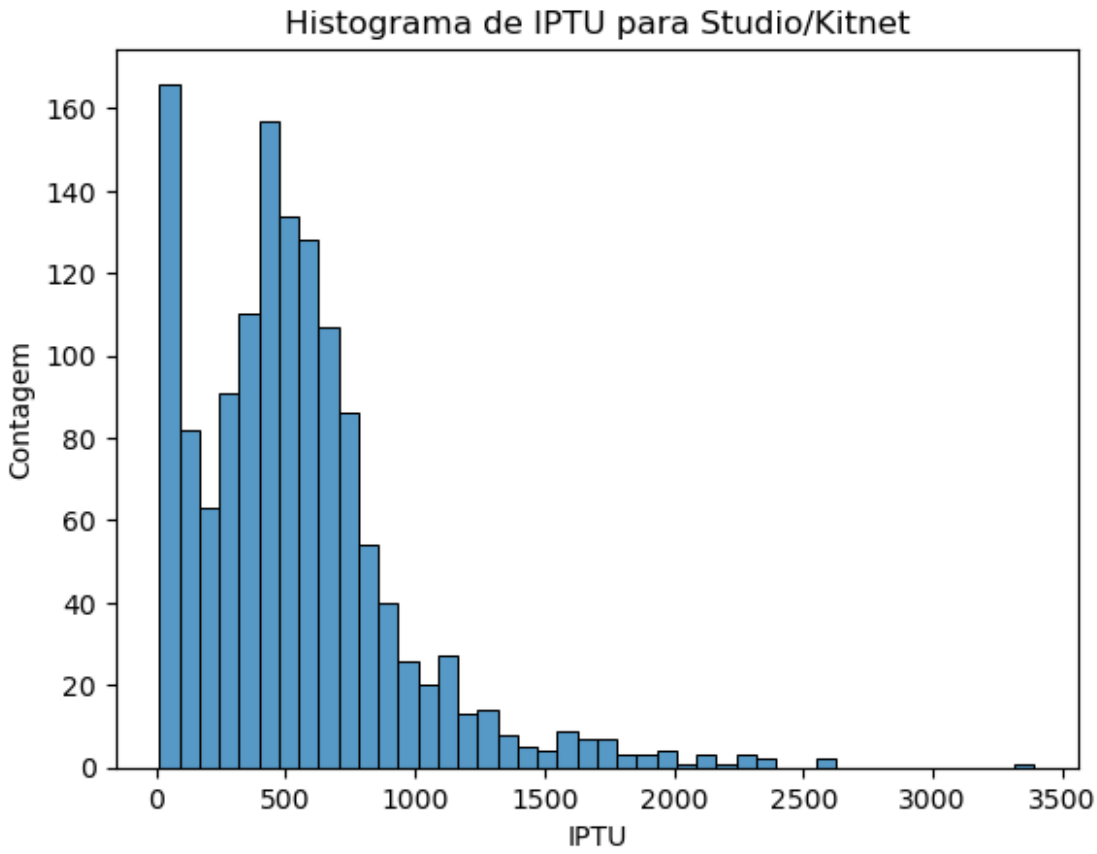
```
for column in colunas_quantitativas:
    sns.histplot(df_studio_kitnet[column])
    plt.title(f'Histograma de {column} para Studio/Kitnet')
    plt.ylabel('Contagem')
    plt.show()
```


Histograma de Area para Studio/Kitnet
Contagem
Area
0
25
50
75
100
125
150
175
200
0
20
40
60
80
100
120

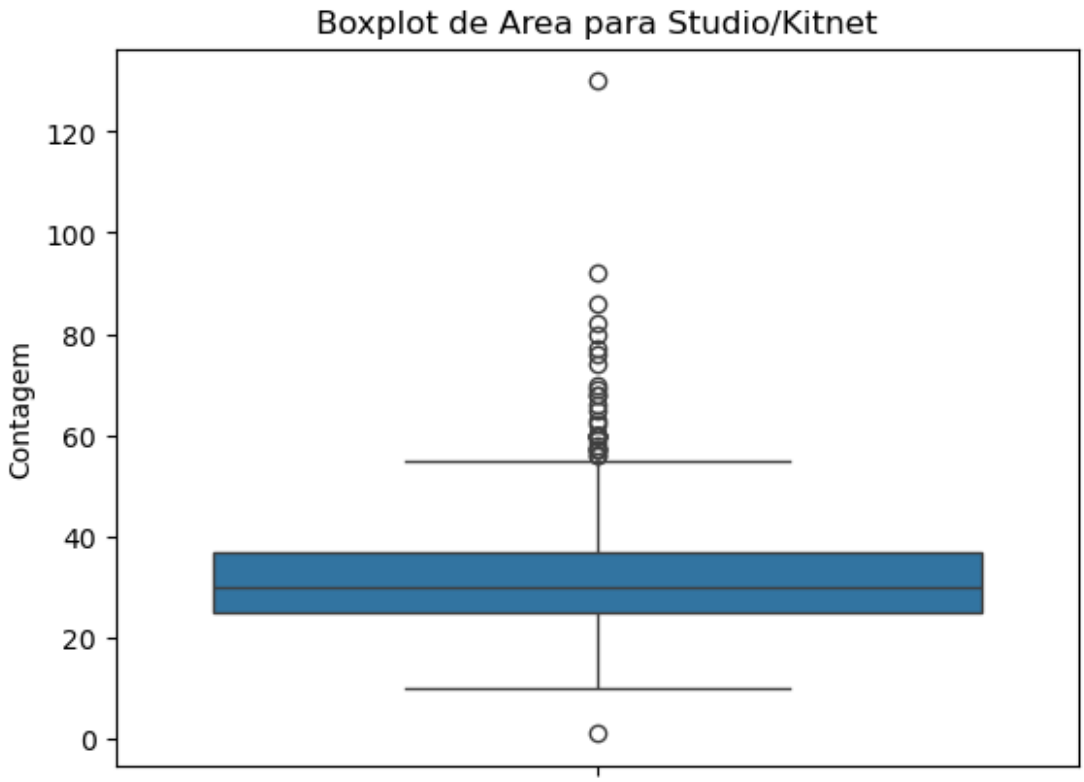

Histograma de Quartos para Studio/Kitnet
Contagem
1200
1000
800
600
400
200
0
0
1
2
3
4
5
6
Quartos


Histograma de Garagem para Studio/Kitnet
Contagem
1000
800
600
400
200
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Garagem

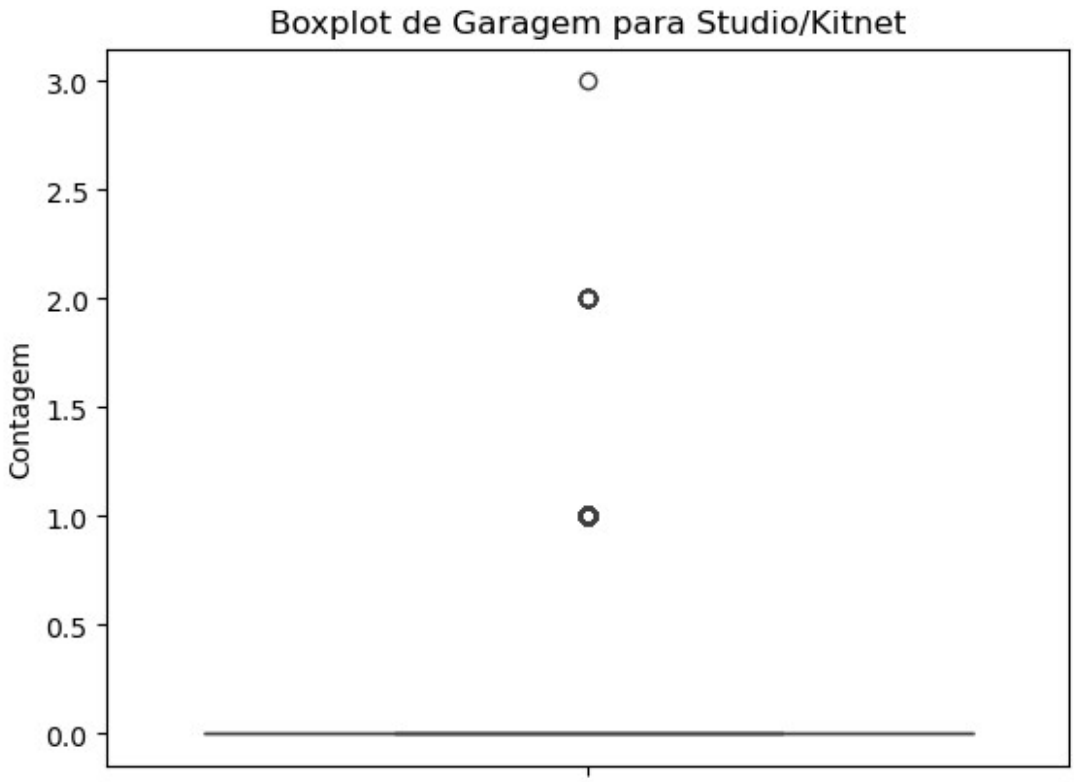

Histograma de Aluguel para Studio/Kitnet
Contagem
Aluguel
0
50
100
150
200
250
0
5000
10000
15000
20000
25000

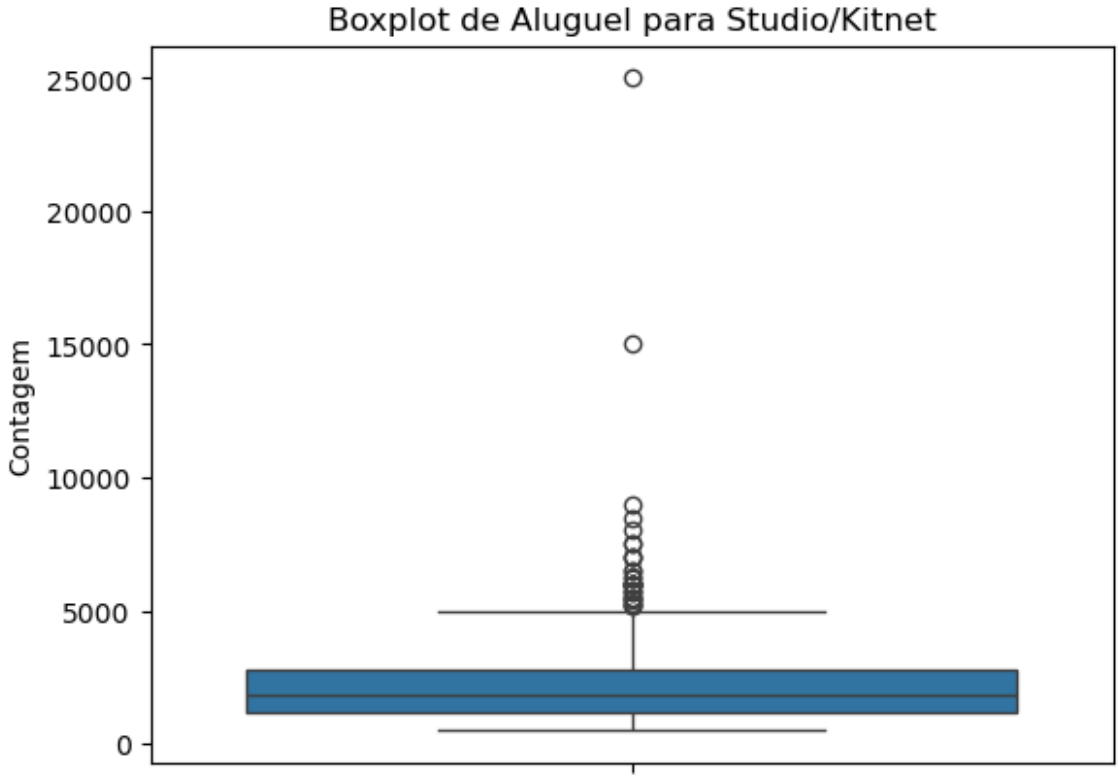


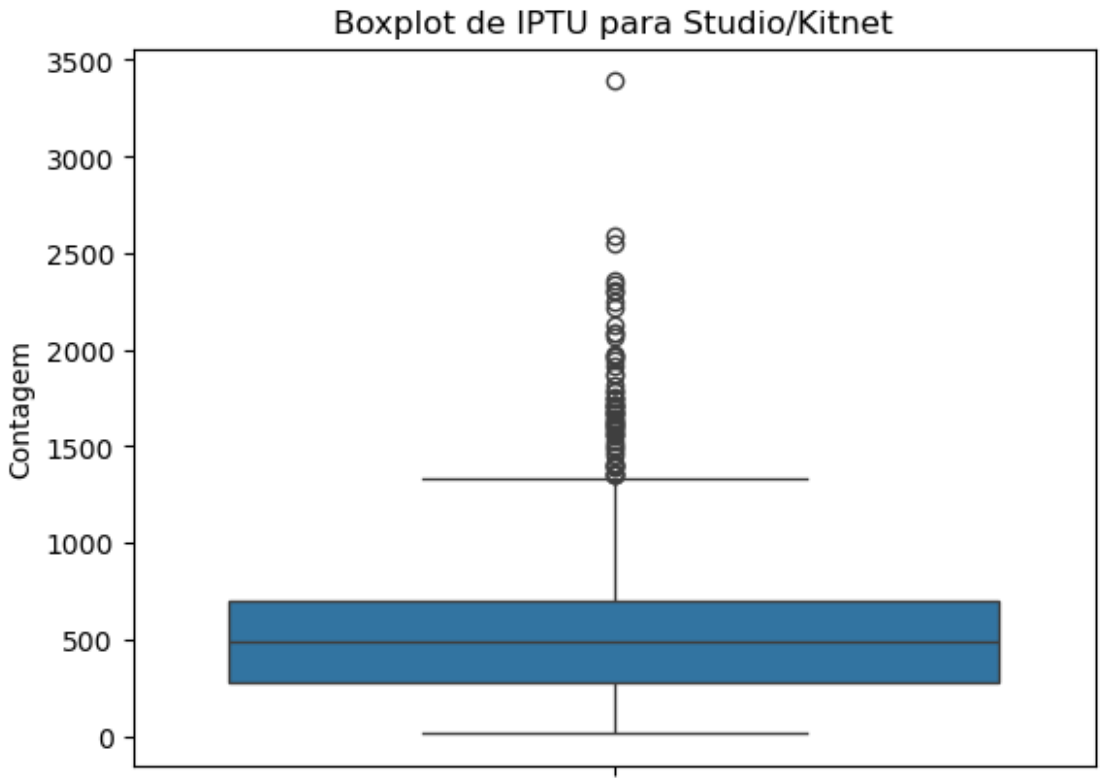

```
for column in colunas_quantitativas:
    sns.boxplot(df_studio_kitnet[column])
    plt.title(f'Boxplot de {column} para Studio/Kitnet')
    plt.ylabel('Contagem')
    plt.show()
```


Boxplot de Area para Studio/Kitnet
Contagem
0
20
40
60
80
100
120


Boxplot de Quartos para Studio/Kitnet
Contagem
0
1
2
3
4
5
6


Boxplot de Garagem para Studio/Kitnet
Contagem
3.0
2.5
2.0
1.5
1.0
0.5
0.0


Boxplot de Aluguel para Studio/Kitnet
Contagem
25000
20000
15000
10000
5000
0




```
df_apartamento = df_bruto.query('Tipo == "Apartamento"')
print(df_apartamento.shape)
df_apartamento.head()
```

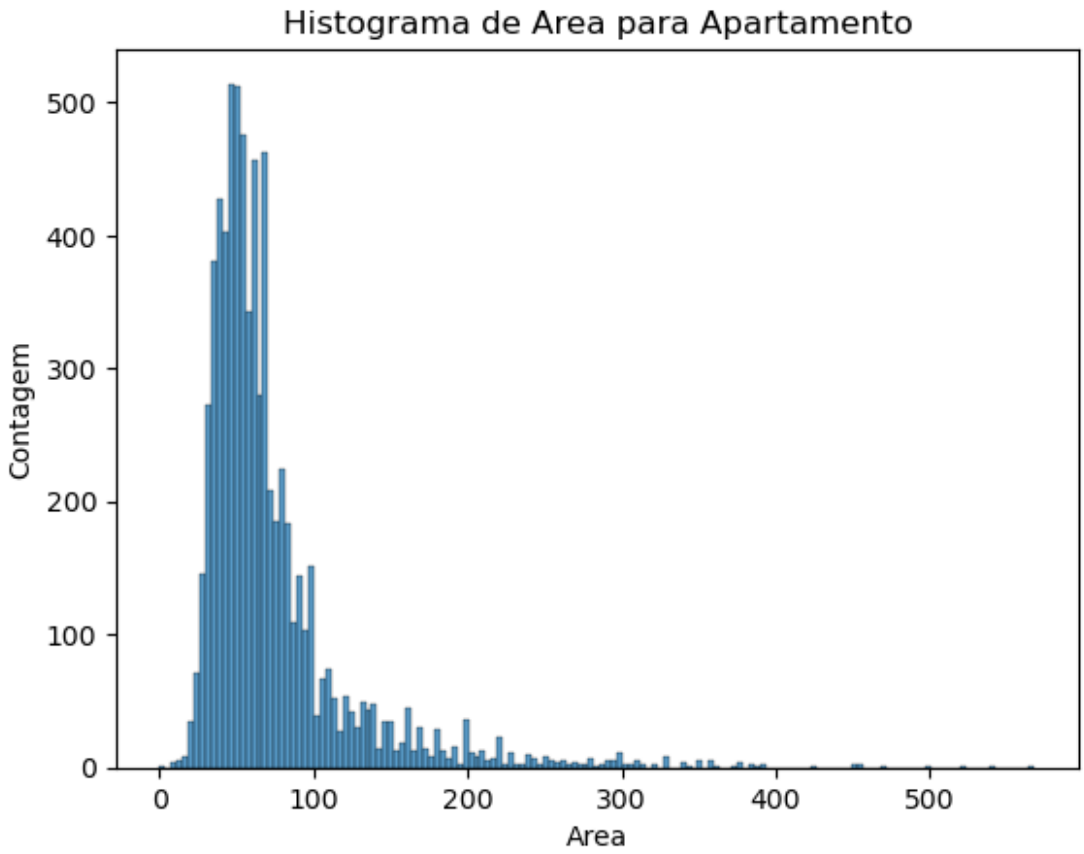
```
(7194, 8)
```

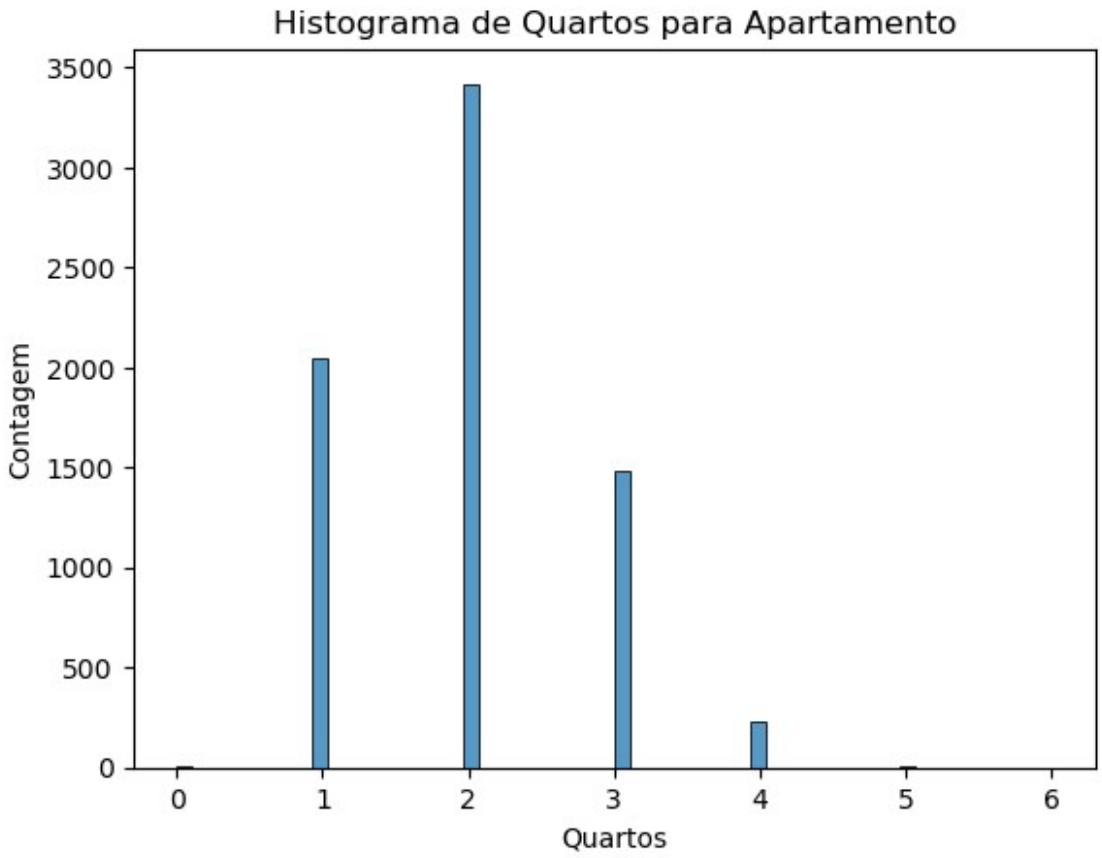
	Endereço	Distrito	Area	Quartos	Garagem	Tipo	Aluguel	IPTU
2	Rua Oscar Freire	Pinheiros	18	1	0	Apartamento	4000	661
5	Rua Domingos Paiva	Brás	50	2	1	Apartamento	3800	787
6	Rua Guararapes	Brooklin Paulista	72	2	1	Apartamento	3500	1687
8	Rua José Peres Campelo	Piqueri	32	2	0	Apartamento	1200	392
9	Rua Guaperuvu	Vila Aricanduva	36	1	0	Apartamento	1200	301

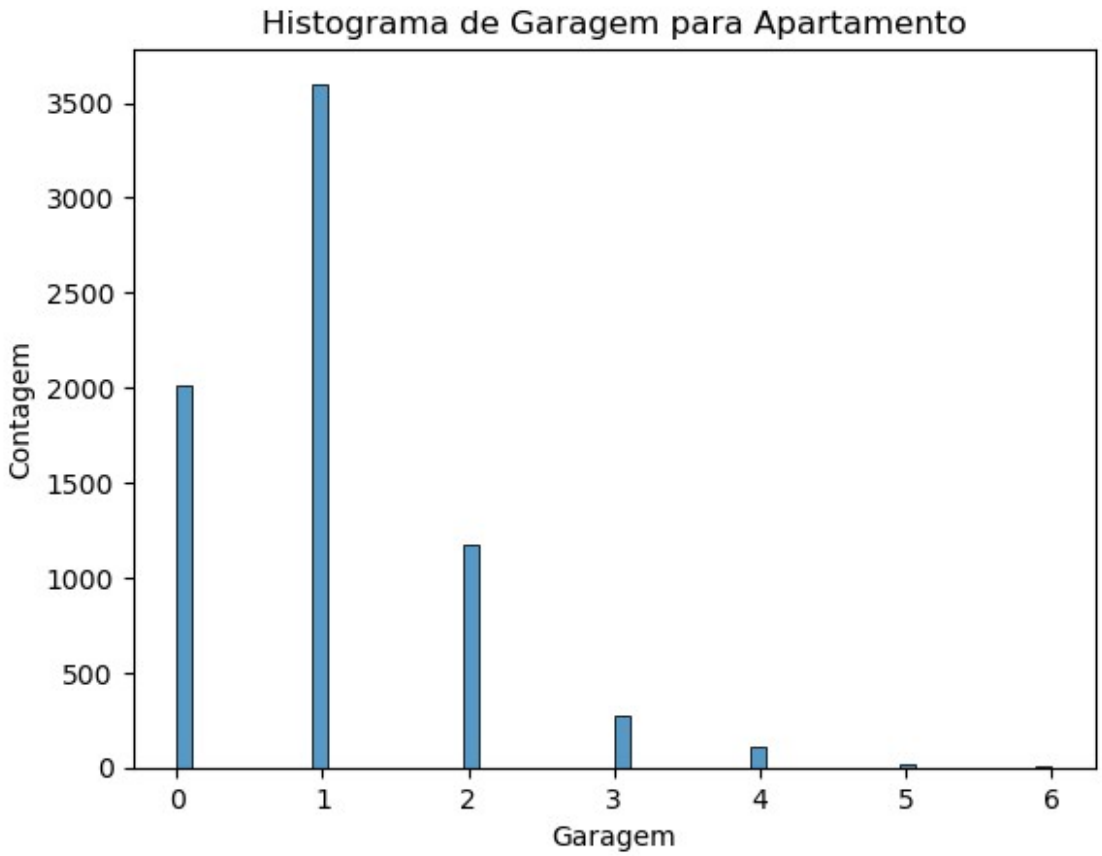
```
df_apartamento.describe()
```

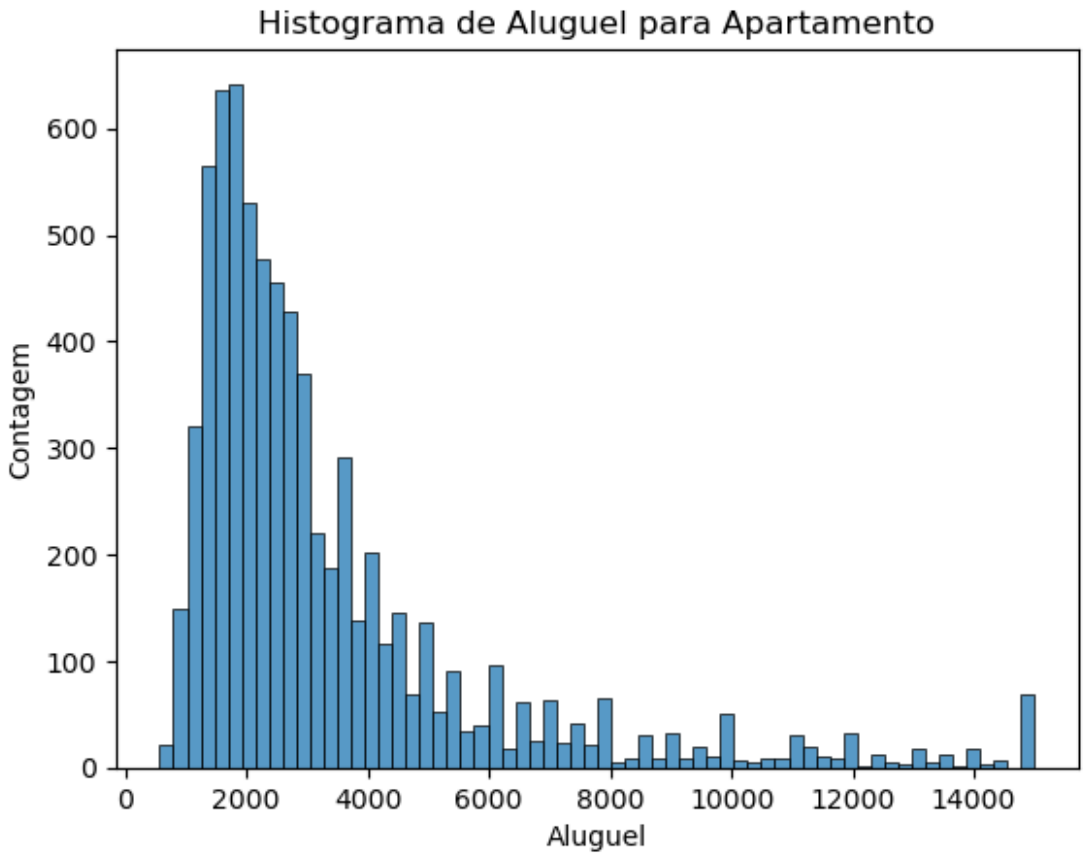
```
              Area      Quartos      Garagem        Aluguel
IPTU
count  7194.000000  7194.000000  7194.000000   7194.000000
7194.000000
mean     73.318460     1.987907     1.022519   3356.902697
1078.525716
std      50.859956     0.798472     0.896132   2638.994113
1049.003501
min       0.000000     0.000000     0.000000    567.000000
13.000000
25%      45.000000     1.000000     0.000000   1700.000000
484.000000
50%      60.000000     2.000000     1.000000   2500.000000
774.500000
75%      81.000000     2.000000     1.000000   3899.750000
1297.000000
max     568.000000     6.000000     6.000000  15000.000000
13700.000000
```

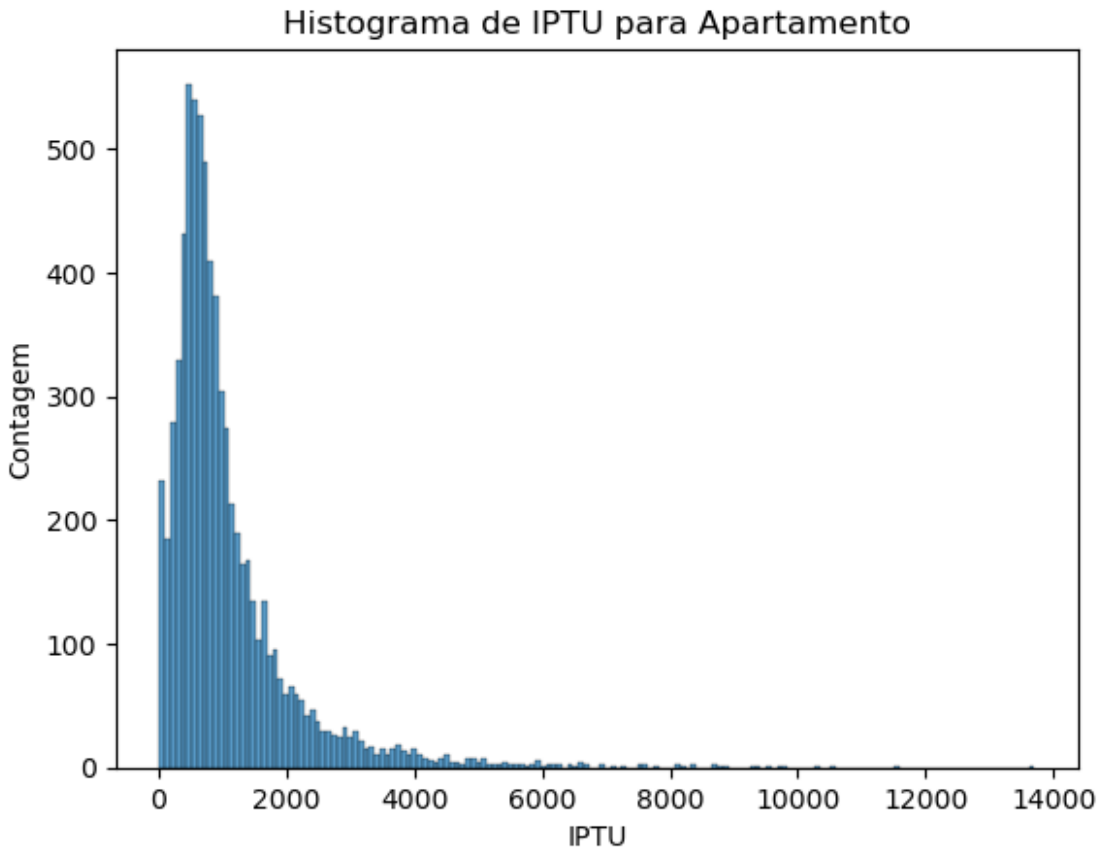
```
for column in colunas_quantitativas:
    sns.histplot(df_apartamento[column])
    plt.title(f'Histograma de {column} para Apartamento')
    plt.ylabel('Contagem')
    plt.show()
```


Histograma de Area para Apartamento
Contagem
Area
0
100
200
300
400
500

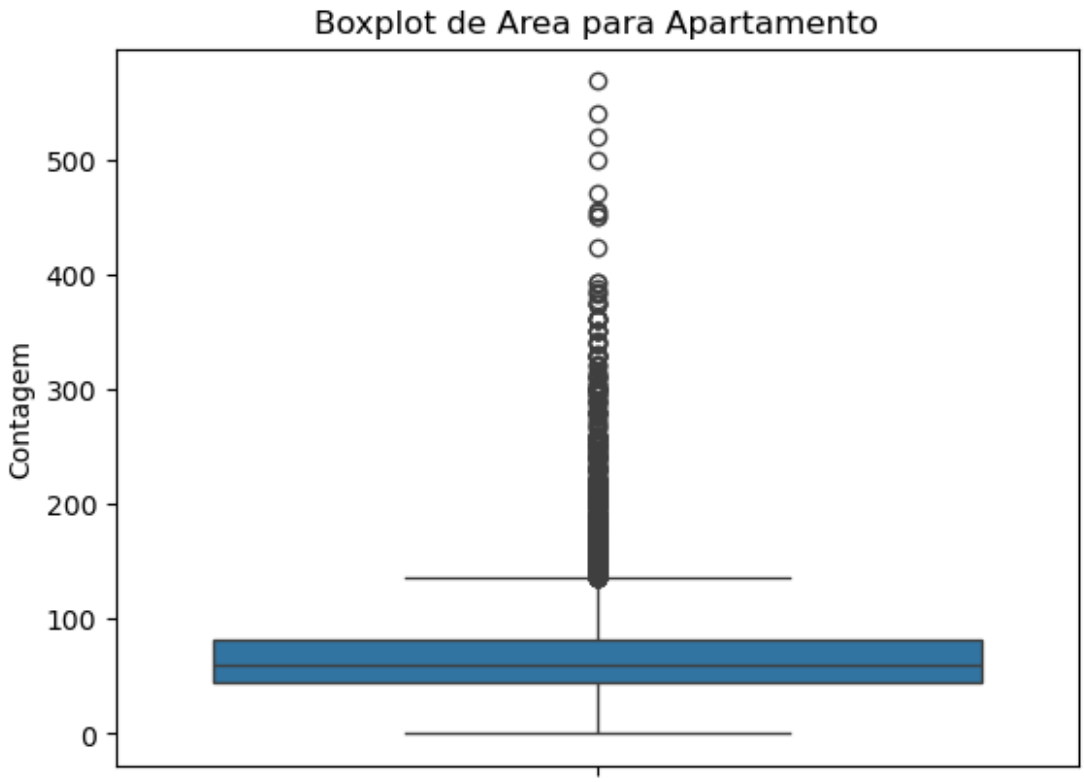

Histograma de Quartos para Apartamento
Contagem
Quartos
0
500
1000
1500
2000
2500
3000
3500
0
1
2
3
4
5
6

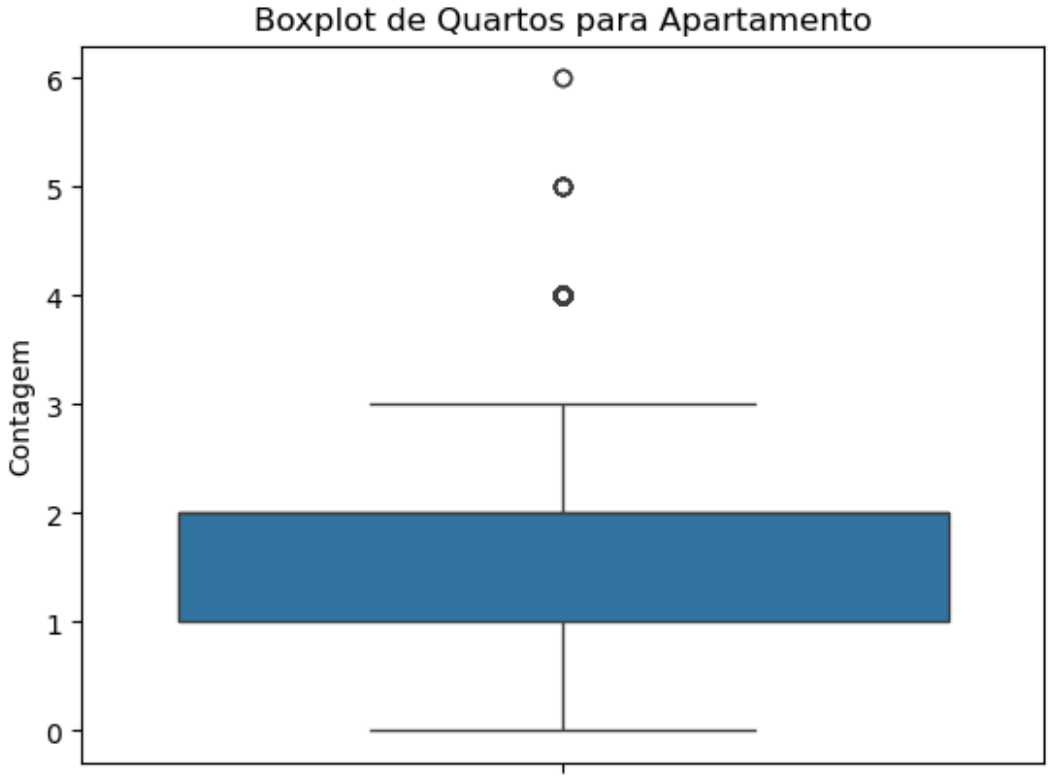

Histograma de Garagem para Apartamento
Contagem
Garagem
0
500
1000
1500
2000
2500
3000
3500
0
1
2
3
4
5
6

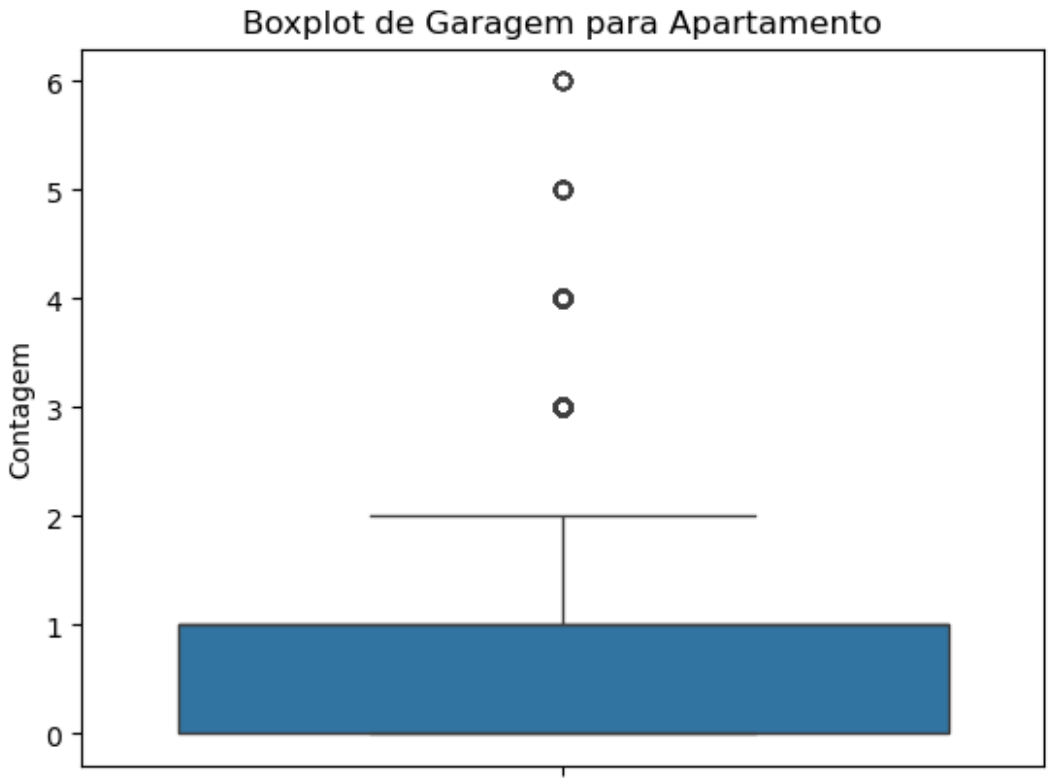

Histograma de Aluguel para Apartamento
Contagem
Aluguel
0
100
200
300
400
500
600
0
2000
4000
6000
8000
10000
12000
14000

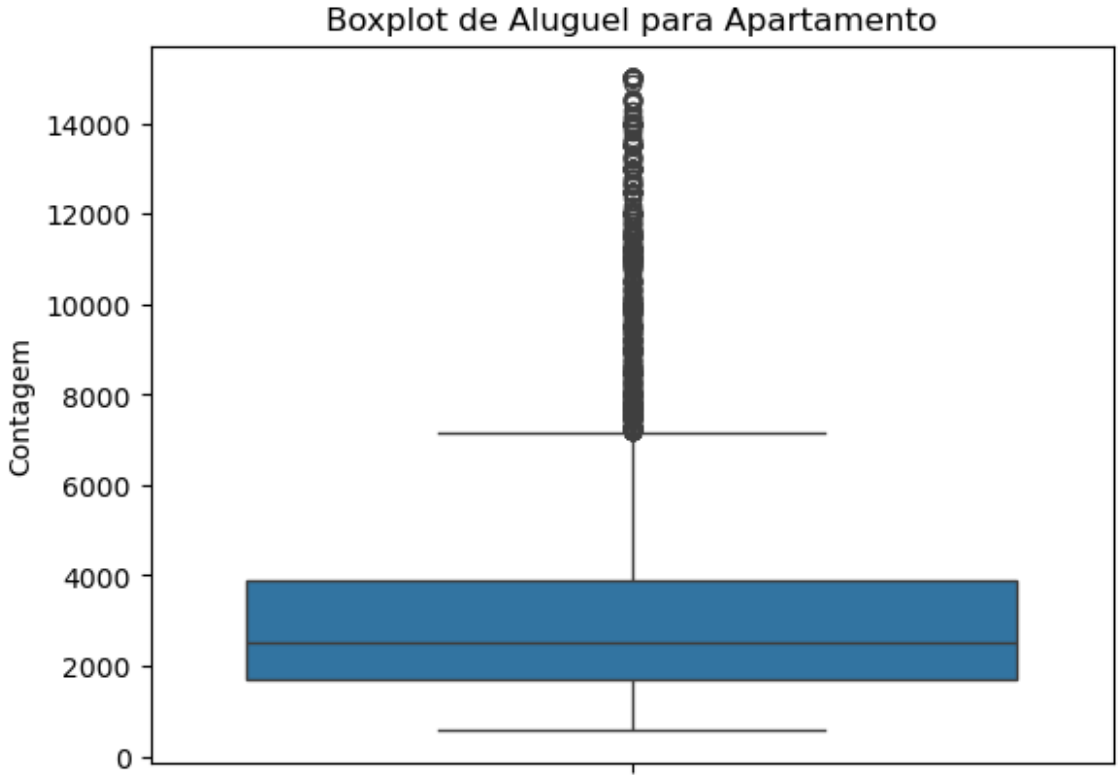


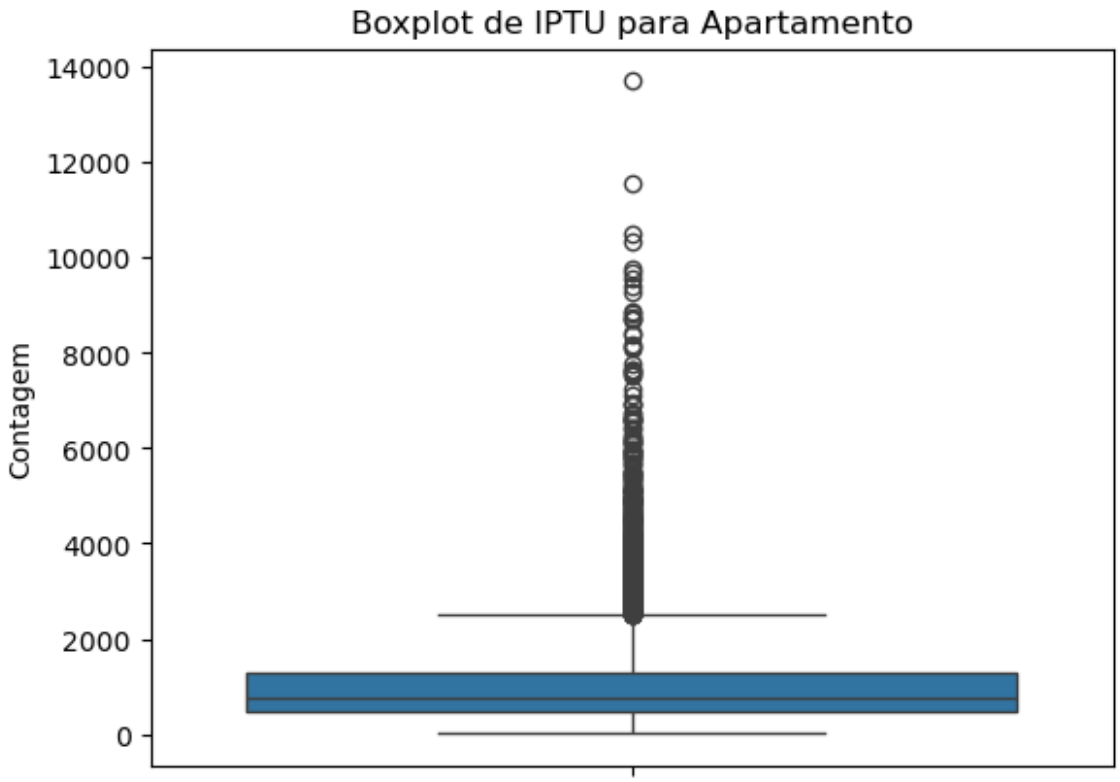

```
for column in colunas_quantitativas:
    sns.boxplot(df_apartamento[column])
    plt.title(f'Boxplot de {column} para Apartamento')
    plt.ylabel('Contagem')
    plt.show()
```


Boxplot de Area para Apartamento
Contagem
0
100
200
300
400
500


Boxplot de Quartos para Apartamento
Contagem
0
1
2
3
4
5
6


Boxplot de Garagem para Apartamento
Contagem
0
1
2
3
4
5
6


Boxplot de Aluguel para Apartamento
Contagem
0
2000
4000
6000
8000
10000
12000
14000




```
df_casa = df_bruto.query('Tipo == "Casa"')
print(df_casa.shape)
df_casa.head()
```

```
(2841, 8)
```

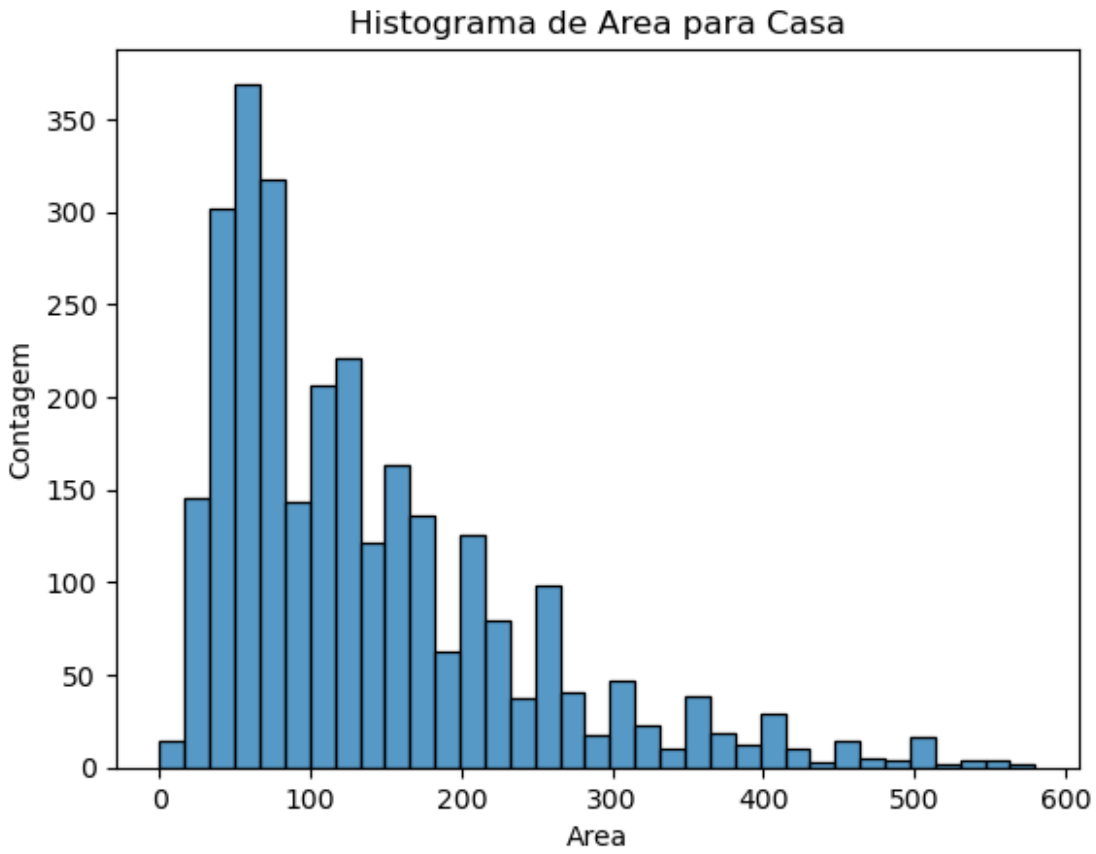
	Endereço	Distrito	Area	Quartos	Garagem	Tipo	\
14	Rua Orestes Barbosa	Jardim Paraventi	70	2	1	Casa	
15	Rua Scuvero	Cambuci	75	2	0	Casa	
18	Rua Guaraiuva	Cidade Monções	30	1	1	Casa	
29	Rua Marcelo Homem de Melo	Quarta Parada	62	2	0	Casa	
32	Rua Nova dos Portugueses	Chora Menino	100	3	2	Casa	

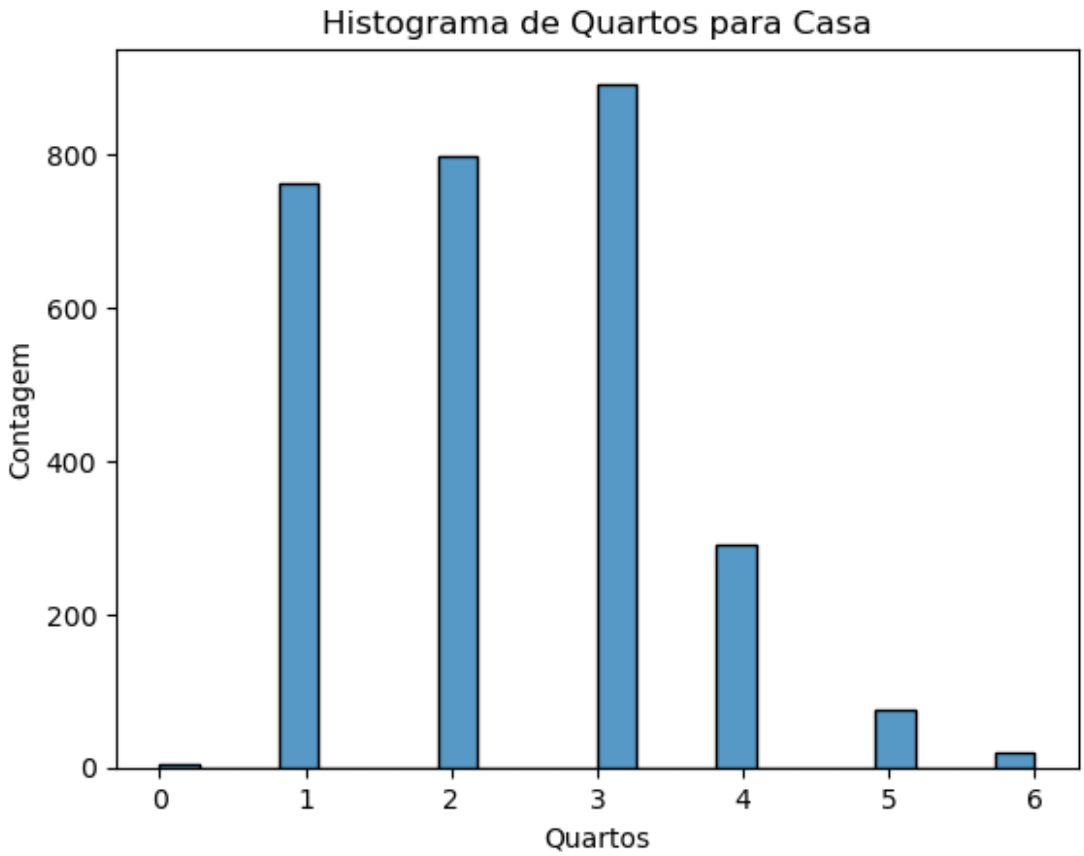
	Aluguel	IPTU
14	1600	168
15	2266	156
18	2394	144
29	2500	94
32	2450	321

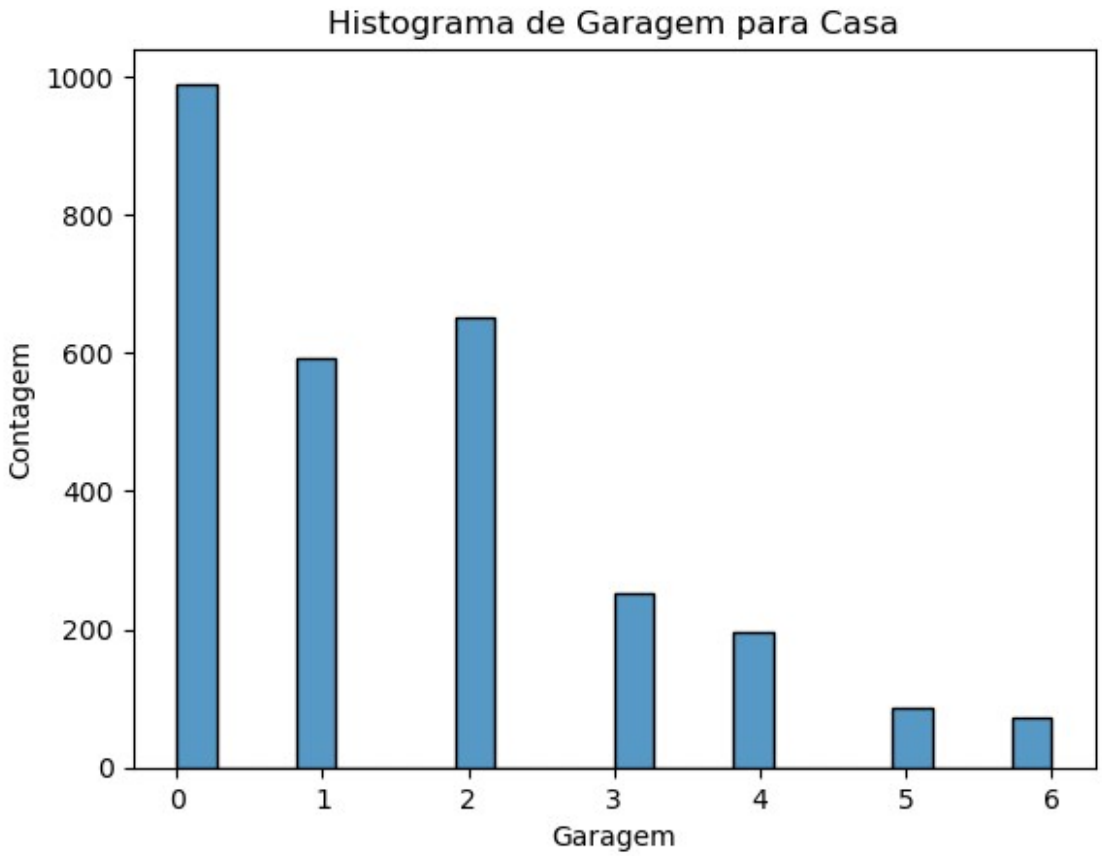
```
df_casa.describe()
```

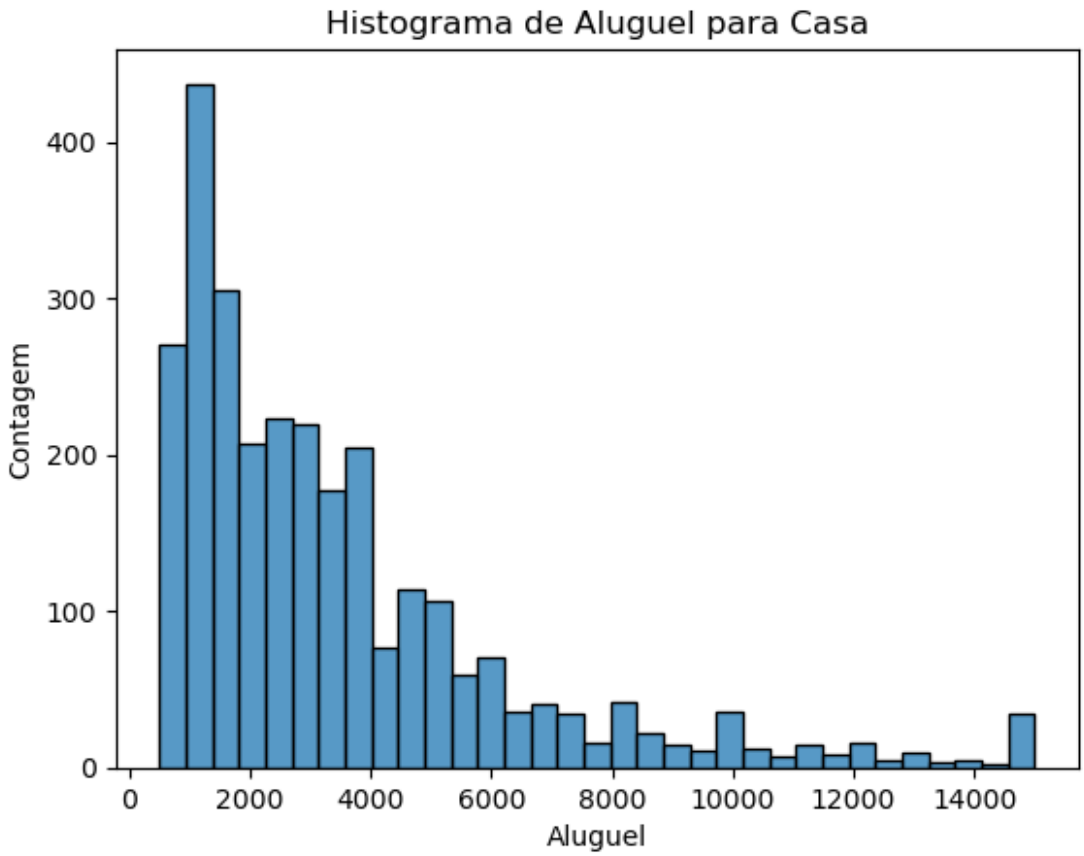
```
              Area      Quartos      Garagem       Aluguel
IPTU
count  2841.000000  2841.000000  2841.000000   2841.000000
2841.000000
mean    136.136220     2.353749     1.514960   3471.924674
352.319606
std     101.794391     1.103369     1.553462   2873.786579
501.148866
min       0.000000     0.000000     0.000000    500.000000
9.000000
25%      60.000000     1.000000     0.000000   1380.000000
97.000000
50%     110.000000     2.000000     1.000000   2600.000000
193.000000
75%     180.000000     3.000000     2.000000   4500.000000
418.000000
max     580.000000     6.000000     6.000000  15000.000000
7380.000000
```

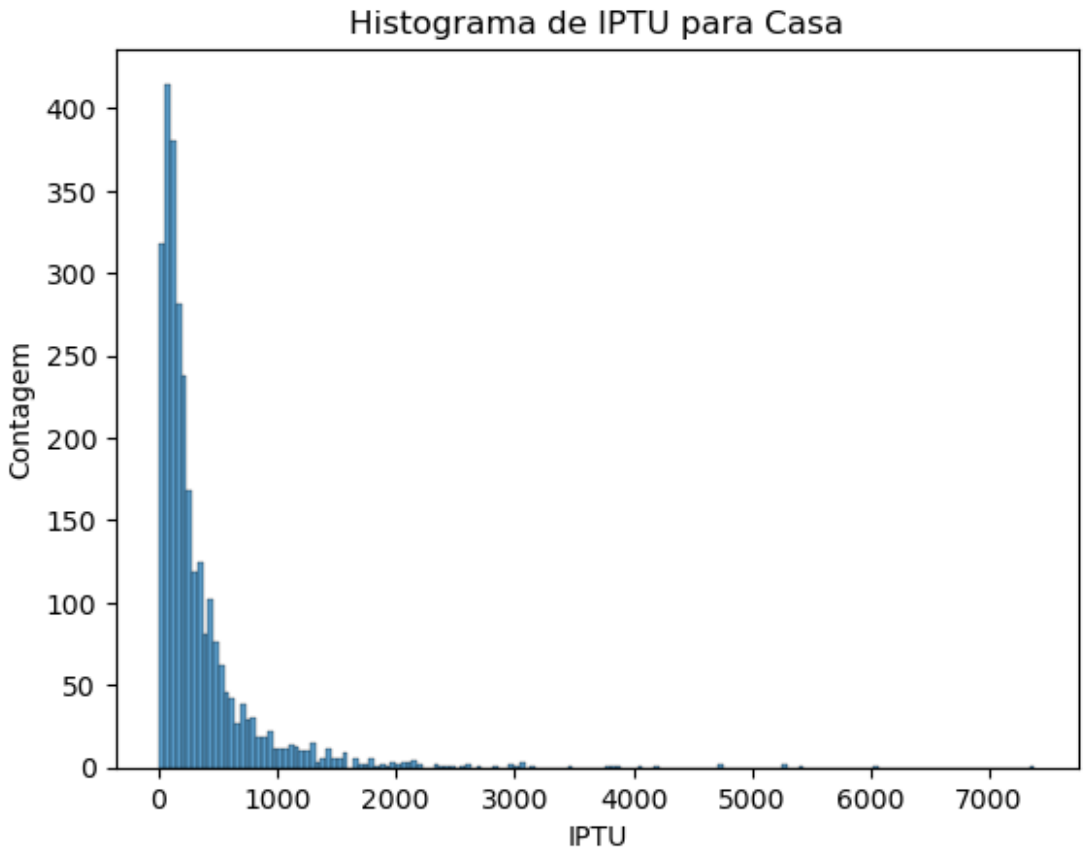
```
for column in colunas_quantitativas:
    sns.histplot(df_casa[column])
    plt.title(f'Histograma de {column} para Casa')
    plt.ylabel('Contagem')
    plt.show()
```


Histograma de Area para Casa
Contagem
Area
0
50
100
150
200
250
300
350
0
100
200
300
400
500
600

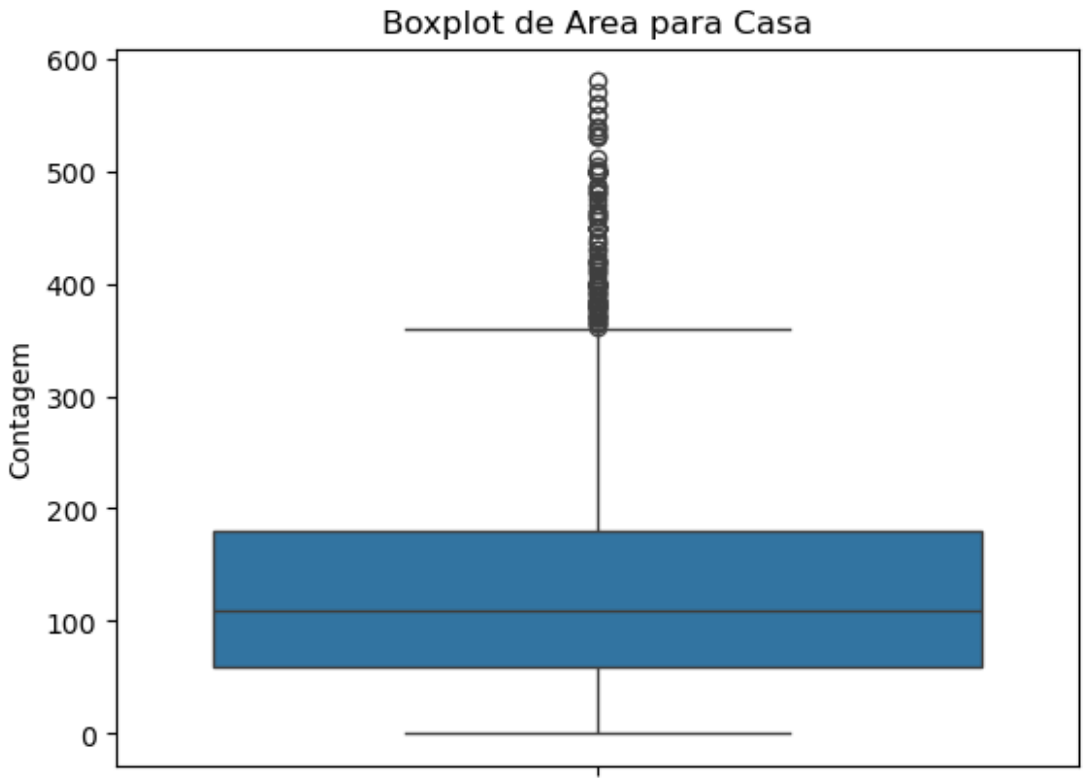

Histograma de Quartos para Casa
Contagem
0
200
400
600
800
0
1
2
3
4
5
6
Quartos

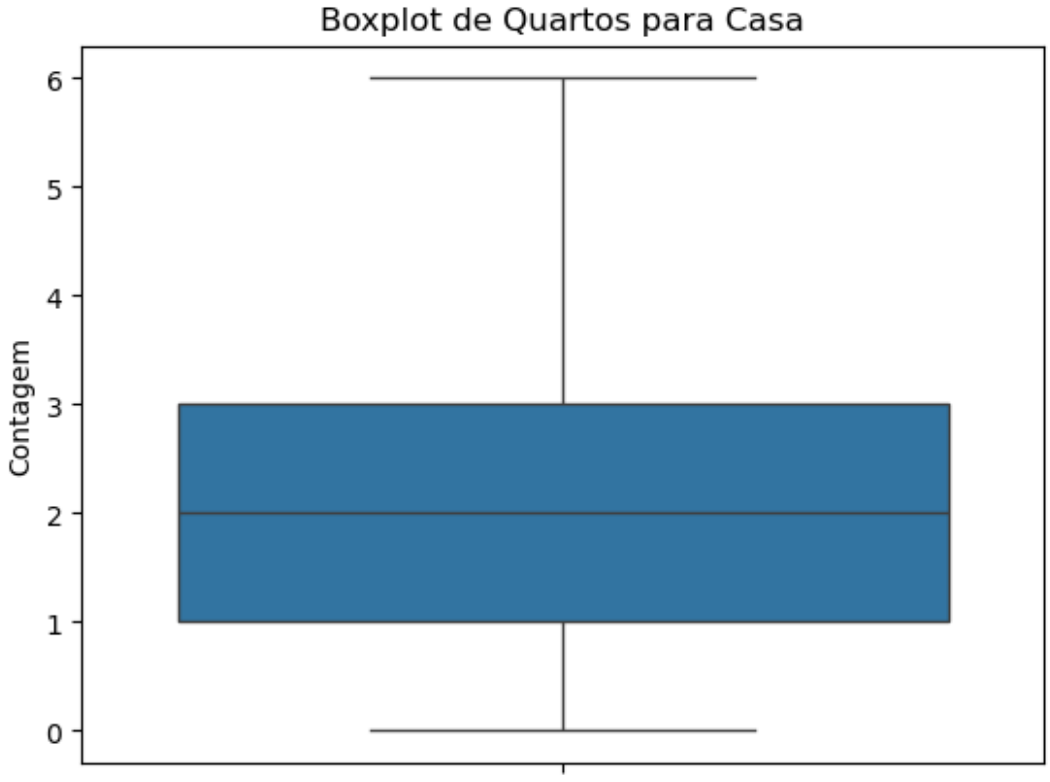

Histograma de Garagem para Casa
1000
800
600
400
200
0
Contagem
0
1
2
3
4
5
6
Garagem

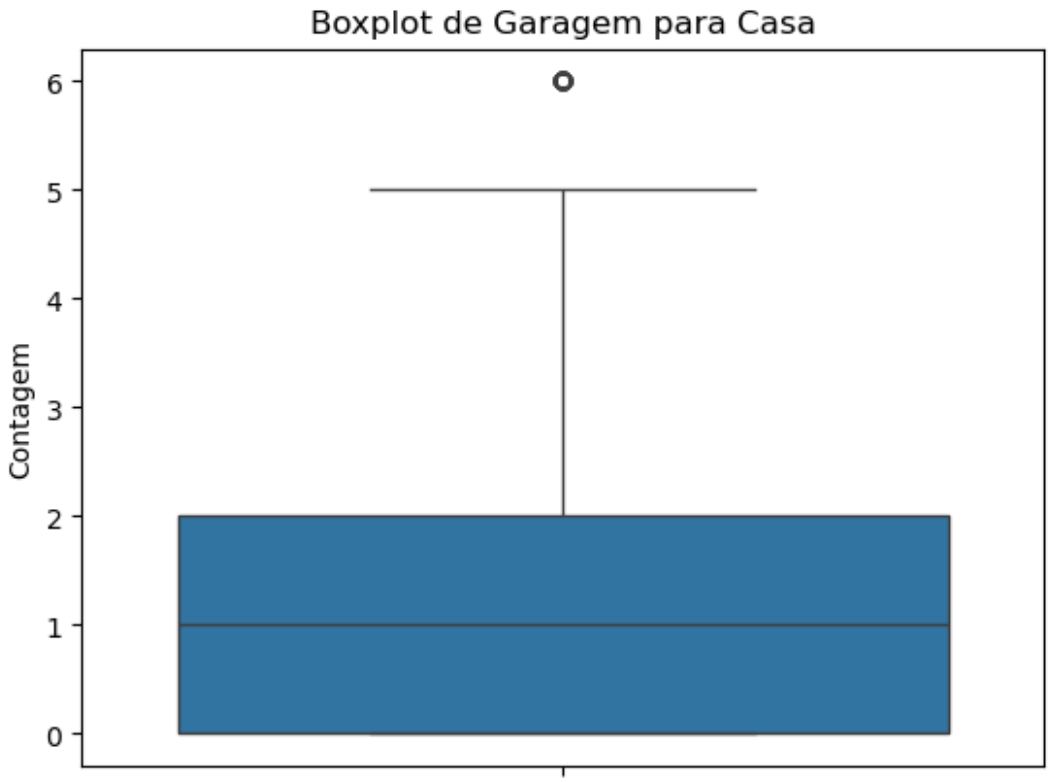

Histograma de Aluguel para Casa
Contagem
Aluguel
0
100
200
300
400
0
2000
4000
6000
8000
10000
12000
14000

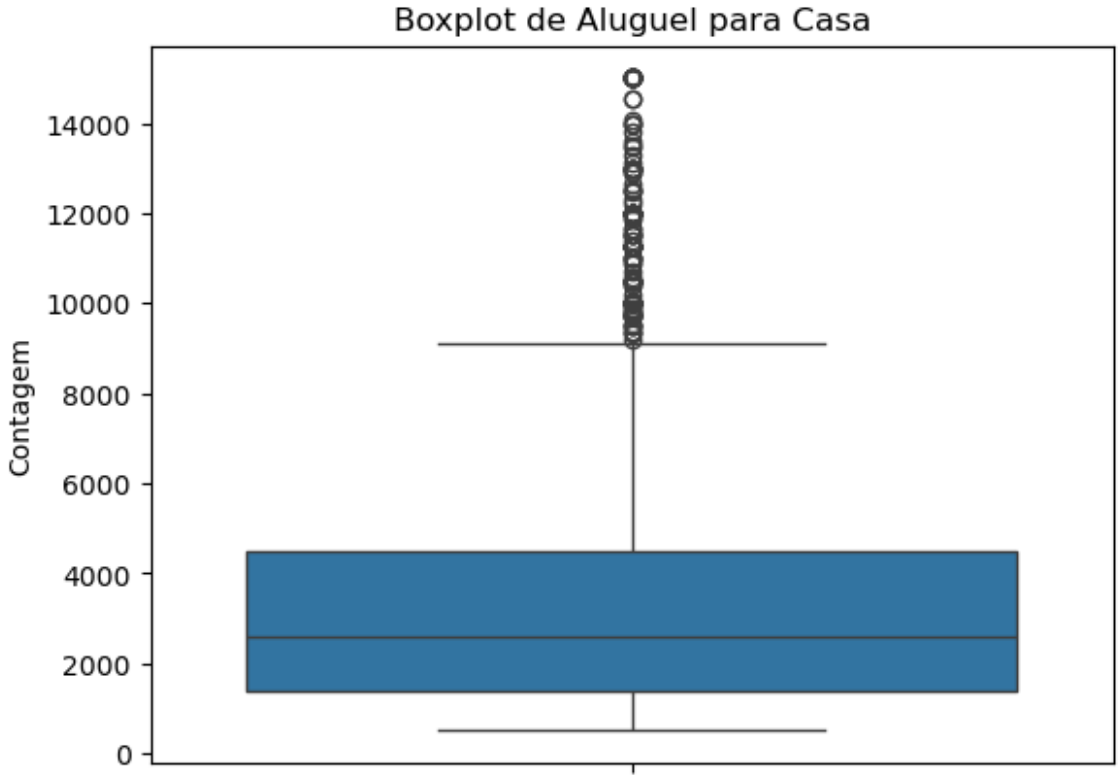


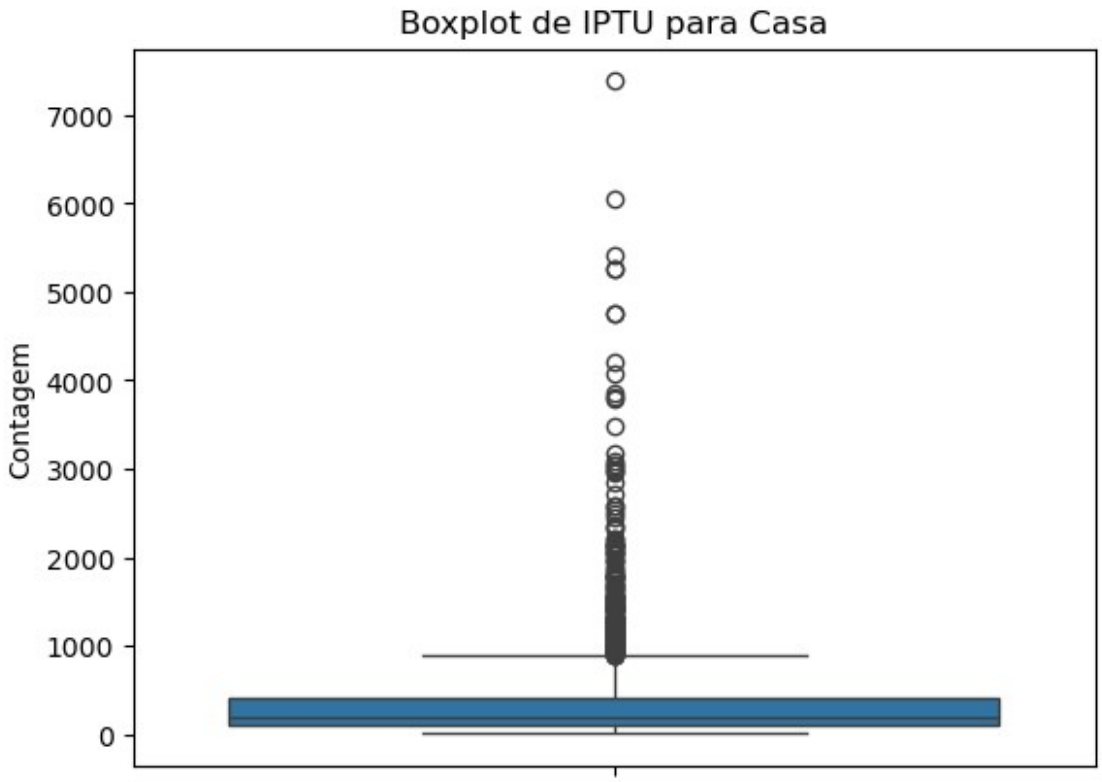

```
for column in colunas_quantitativas:
    sns.boxplot(df_casa[column])
    plt.title(f'Boxplot de {column} para Casa')
    plt.ylabel('Contagem')
    plt.show()
```


Boxplot de Area para Casa
Contagem
600
500
400
300
200
100
0


Boxplot de Quartos para Casa
Contagem
6
5
4
3
2
1
0


Boxplot de Garagem para Casa
Contagem
0
1
2
3
4
5
6


Boxplot de Aluguel para Casa
Contagem
0
2000
4000
6000
8000
10000
12000
14000




```
df_casa_cond = df_bruto.query('Tipo == "Casa em condomínio"')
print(df_casa_cond.shape)
df_casa_cond.head()
```

```
(241, 8)
```

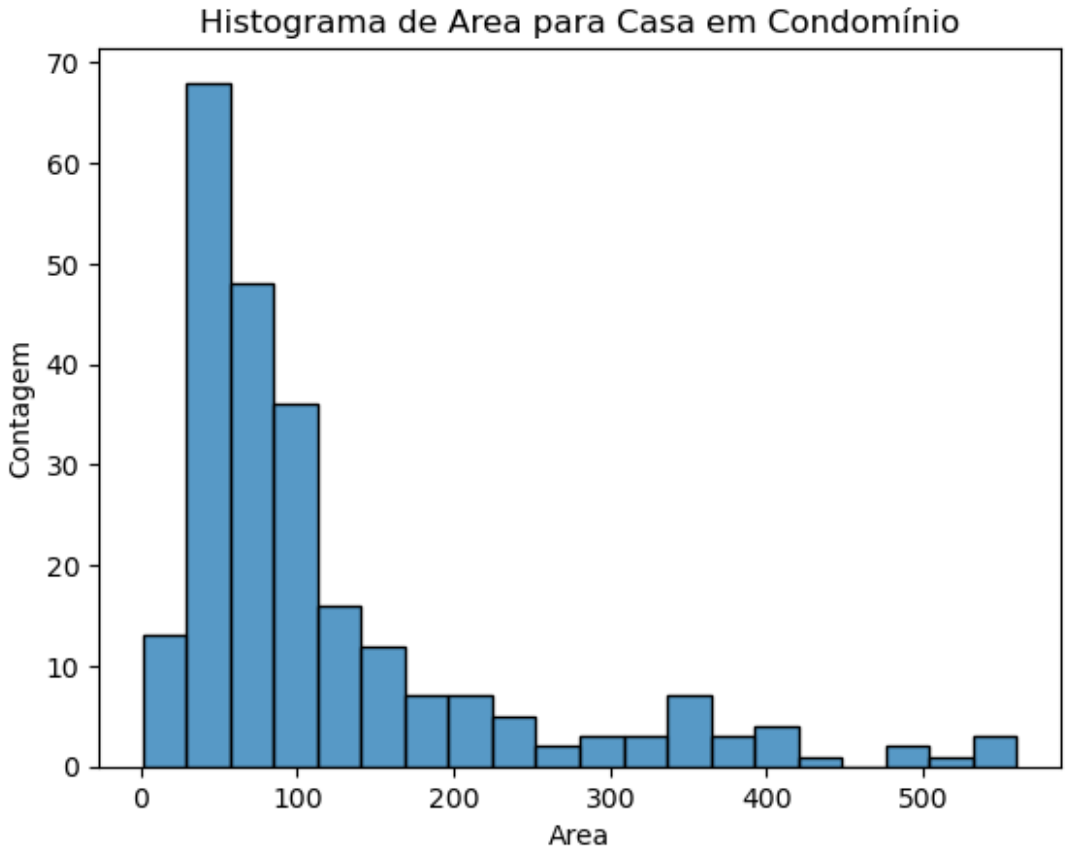
	Endereço	Distrito	Area	Quartos	Garagem	\
3	Rua Júlio Sayago	Vila Ré	56	2	2	
13	Rua Herison	Lauzane Paulista	50	3	0	
89	Rua Tanque Velho	Vila Nivi	42	2	0	
96	Rua Afonso Morsch	Vila Constança	75	2	1	
215	Avenida Francisco Rodrigues	Vila Constança	64	2	1	

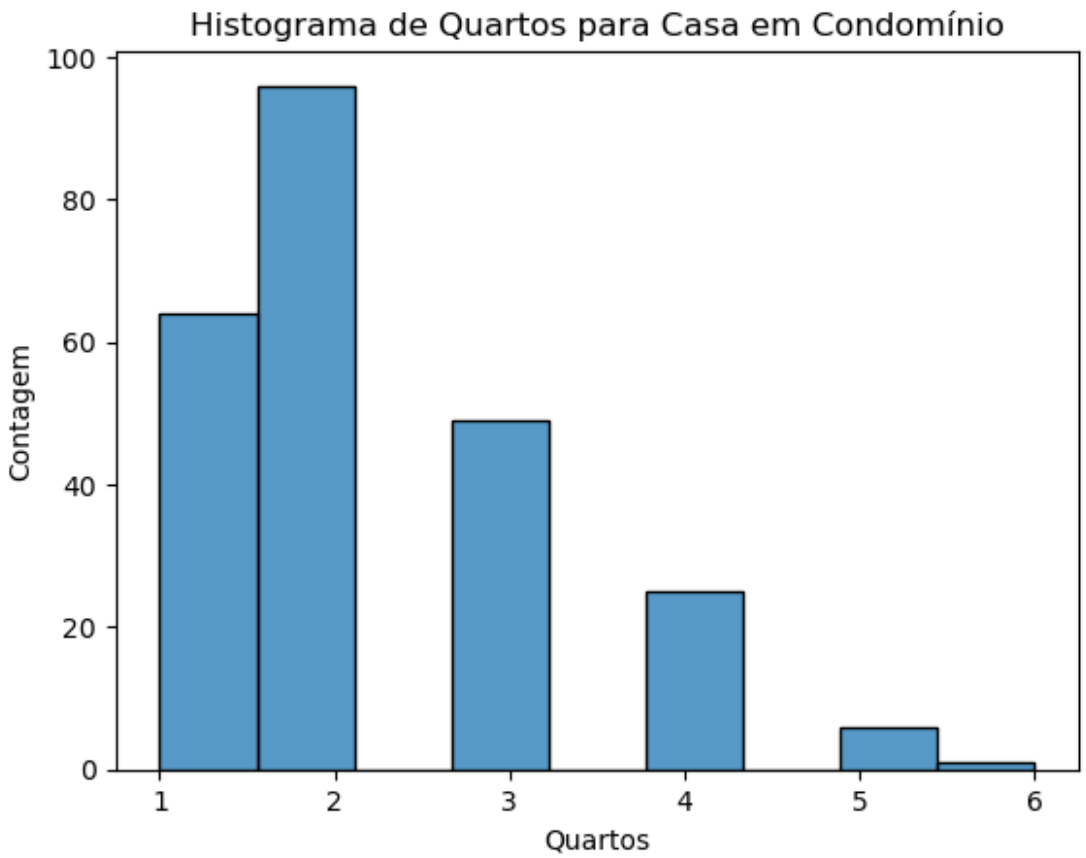
	Tipo	Aluguel	IPTU
3	Casa em condomínio	1750	204
13	Casa em condomínio	1437	80
89	Casa em condomínio	1500	125
96	Casa em condomínio	4000	67
215	Casa em condomínio	2200	213

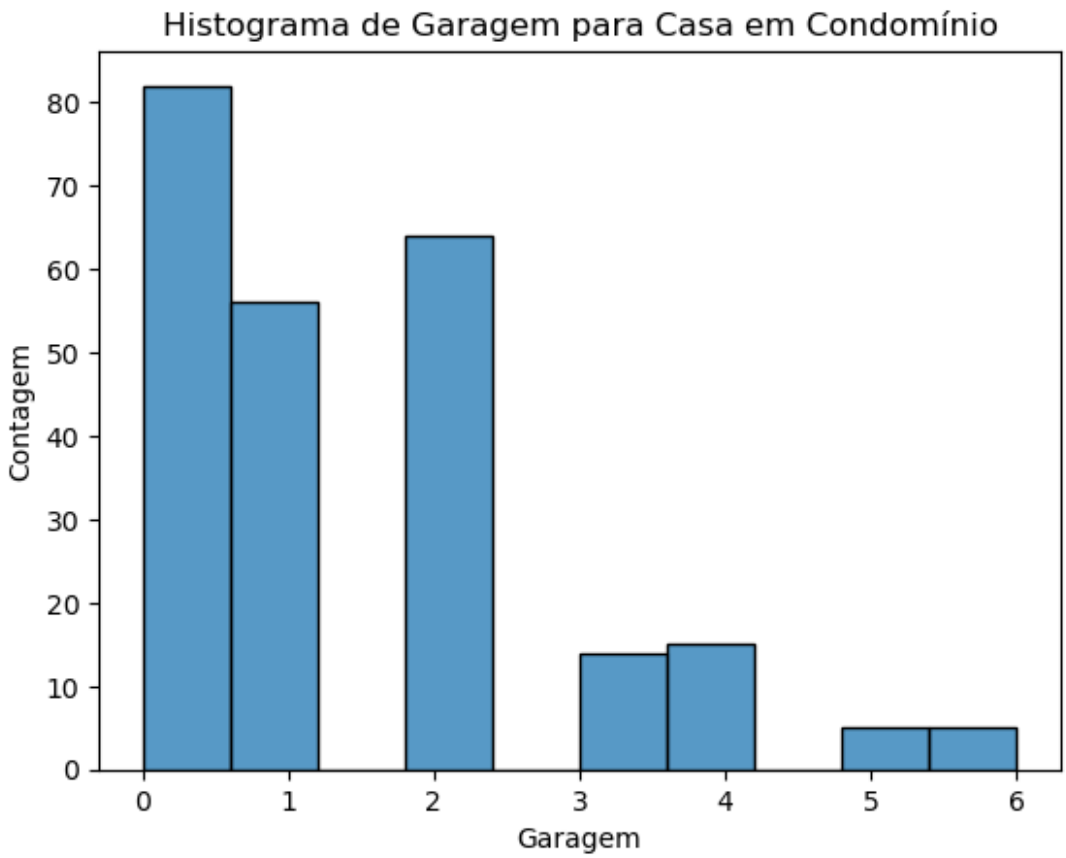
```
df_casa_cond.describe()
```

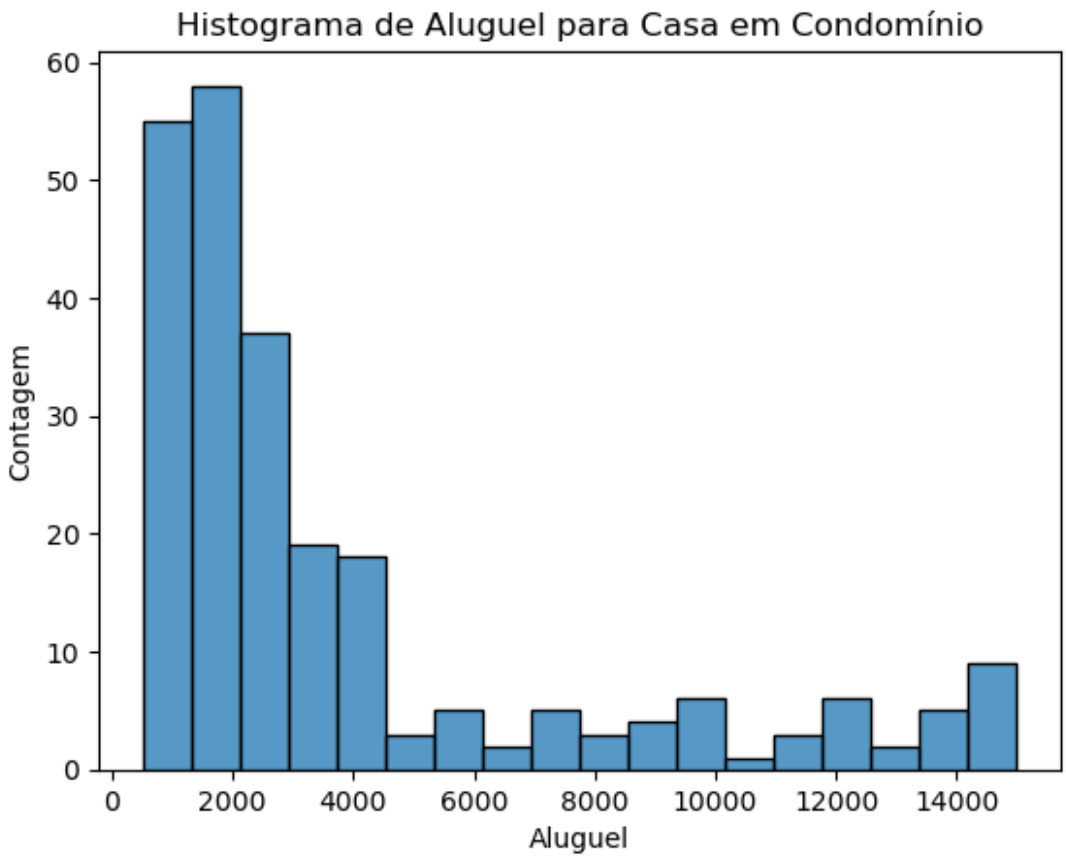
	Area	Quartos	Garagem	Aluguel	IPTU
count	241.000000	241.000000	241.000000	241.000000	241.000000
mean	119.414938	2.236515	1.414938	3912.551867	663.680498
std	113.205206	1.059714	1.444106	3930.906662	930.247051
min	1.000000	1.000000	0.000000	504.000000	22.000000
25%	50.000000	1.000000	0.000000	1400.000000	124.000000
50%	75.000000	2.000000	1.000000	2200.000000	307.000000
75%	140.000000	3.000000	2.000000	4000.000000	712.000000
max	560.000000	6.000000	6.000000	15000.000000	6140.000000

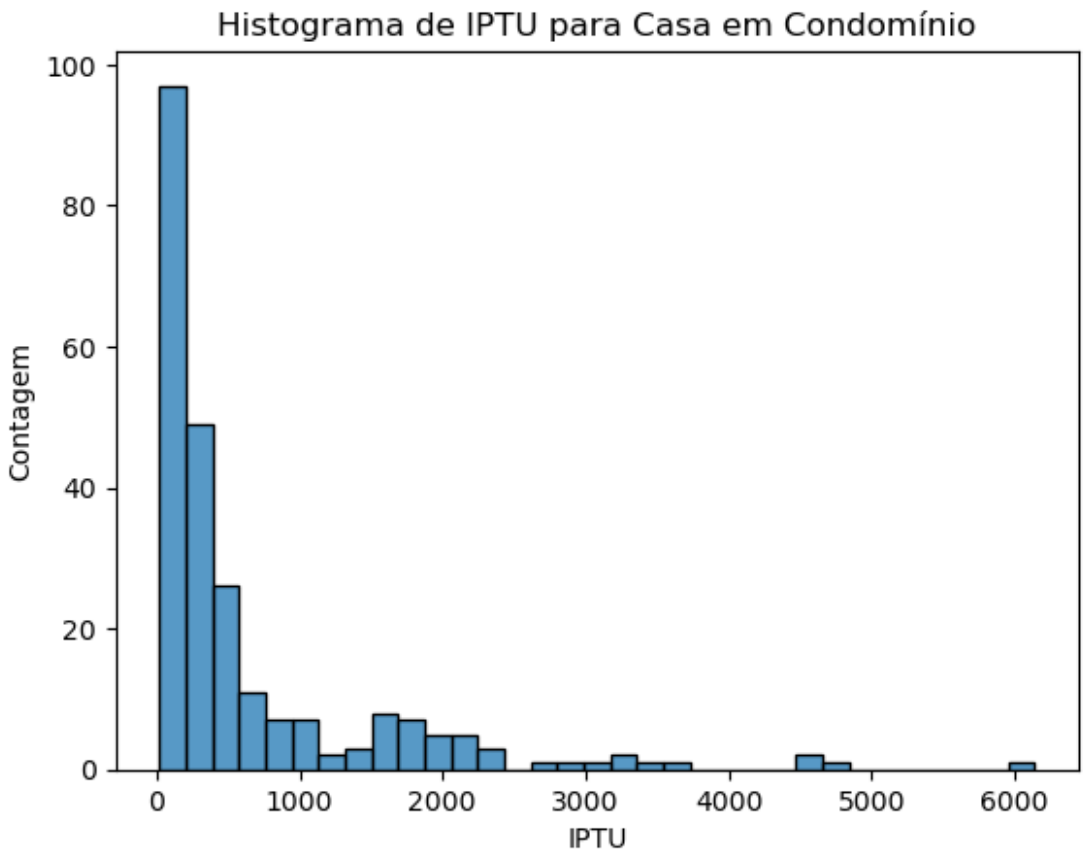
```
for column in colunas_quantitativas:
    sns.histplot(df_casa_cond[column])
    plt.title(f'Histograma de {column} para Casa em Condomínio')
    plt.ylabel('Contagem')
    plt.show()
```



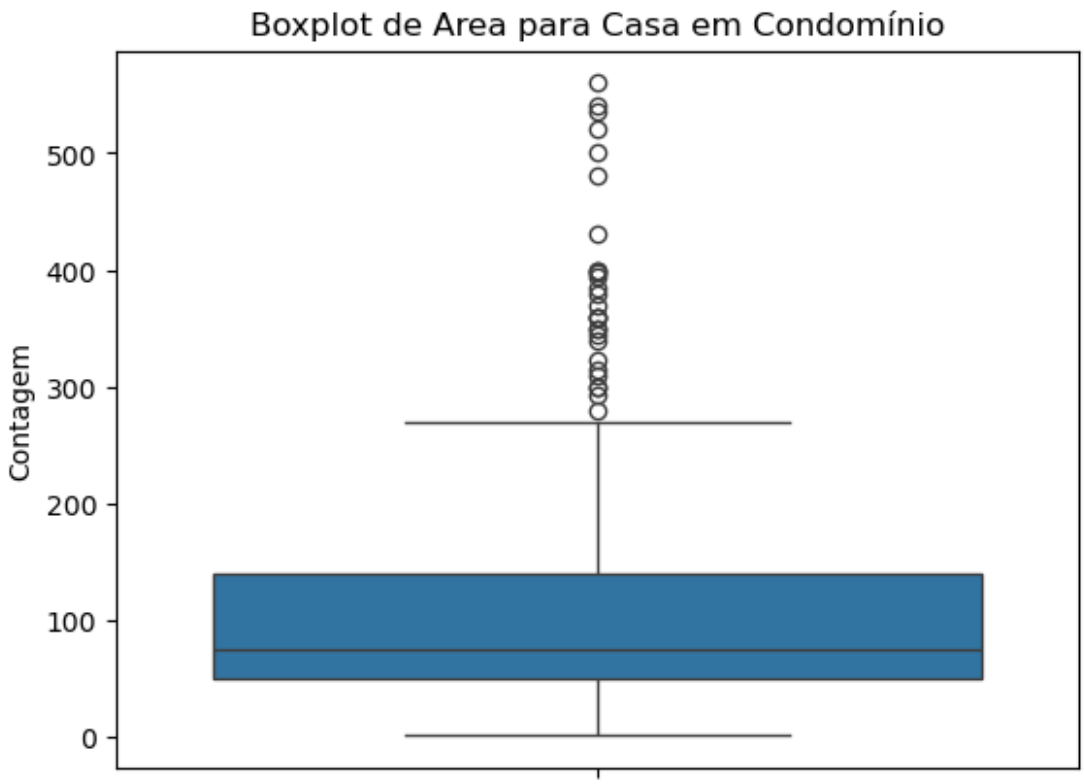

Histograma de Quartos para Casa em Condomínio
Contagem
Quartos
0
20
40
60
80
100
1
2
3
4
5
6

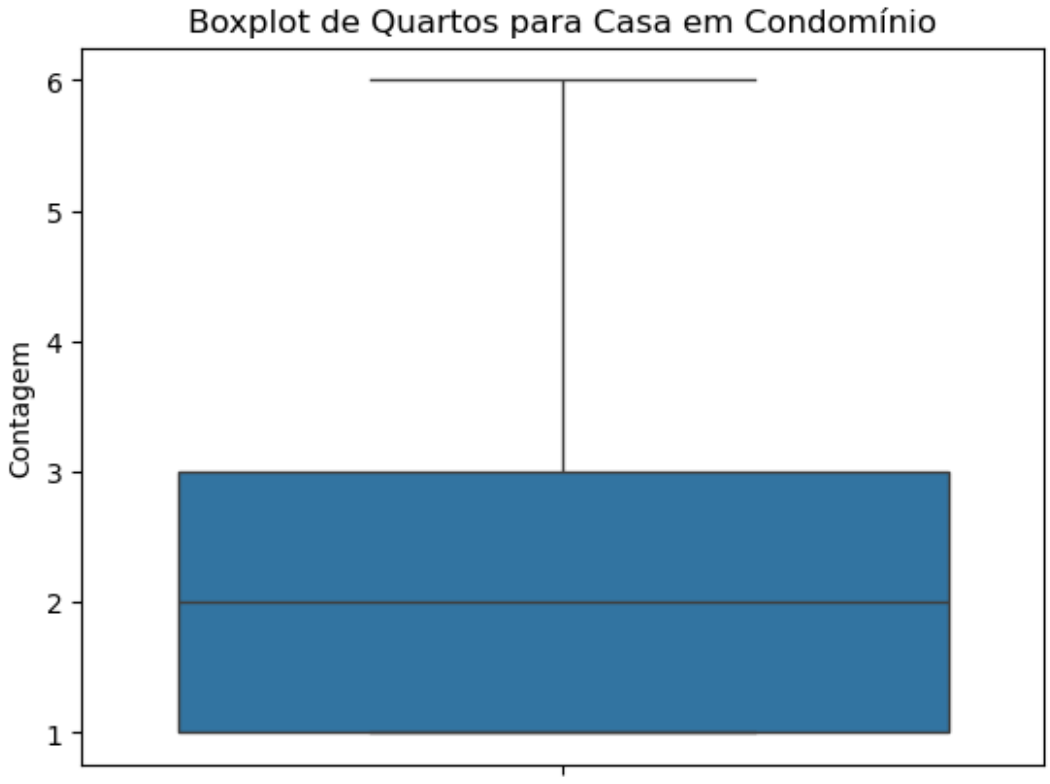

Histograma de Garagem para Casa em Condomínio
Contagem
Garagem
0
10
20
30
40
50
60
70
80
0
1
2
3
4
5
6

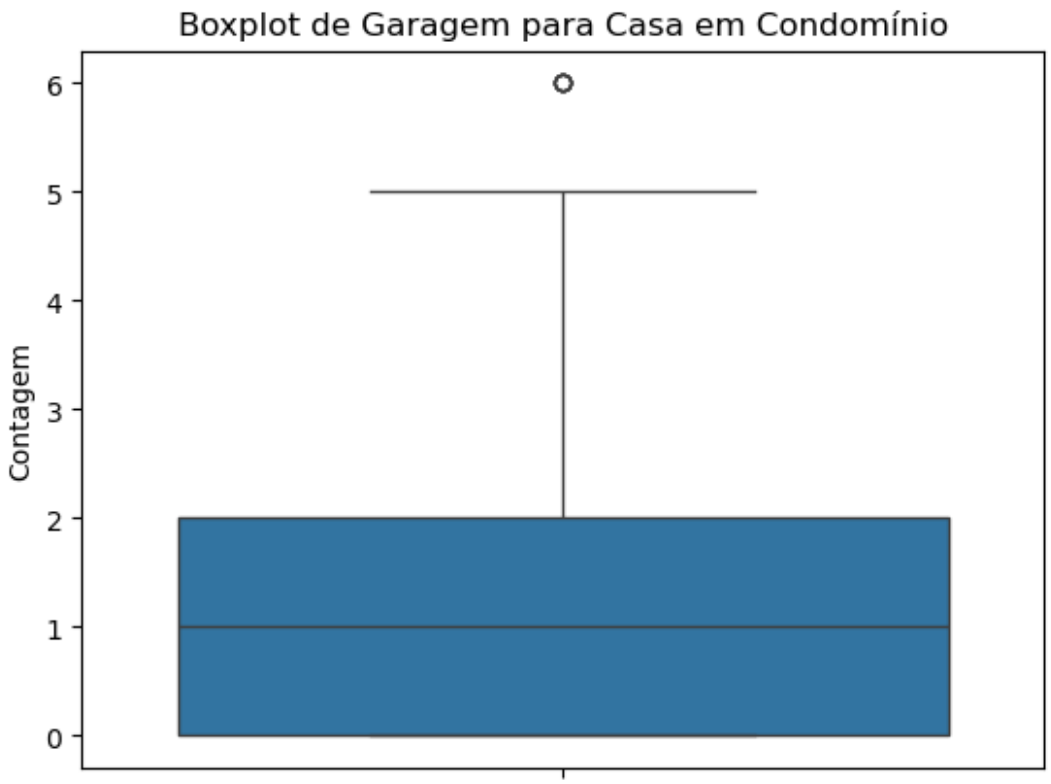

Histograma de Aluguel para Casa em Condomínio
Contagem
Aluguel
0
10
20
30
40
50
60
0
2000
4000
6000
8000
10000
12000
14000

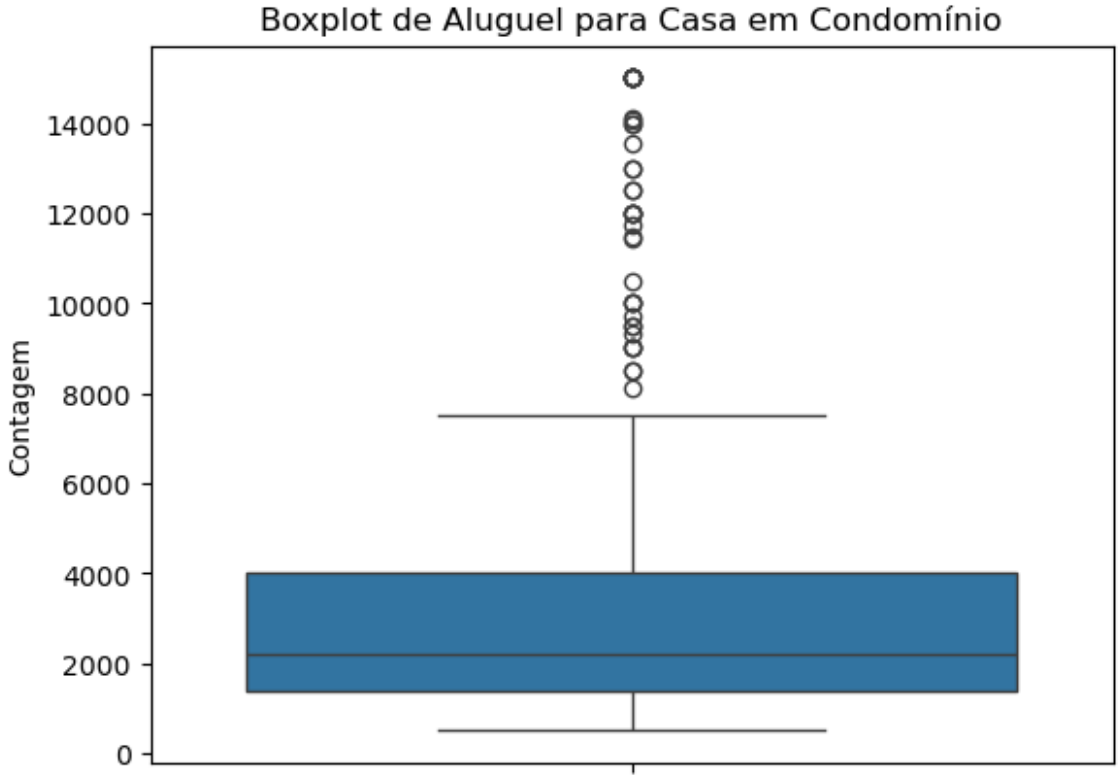


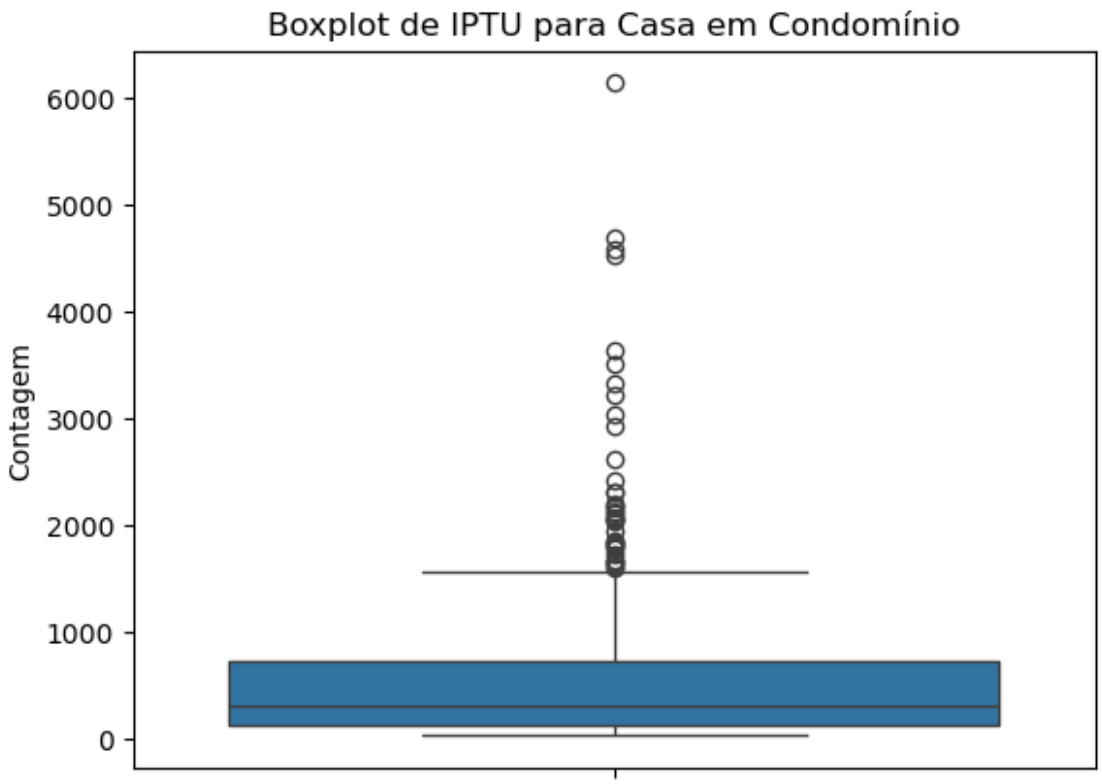

```
for column in colunas_quantitativas:
    sns.boxplot(df_casa_cond[column])
    plt.title(f'Boxplot de {column} para Casa em Condomínio')
    plt.ylabel('Contagem')
    plt.show()
```


Boxplot de Area para Casa em Condomínio
Contagem
0
100
200
300
400
500


Boxplot de Quartos para Casa em Condomínio
Contagem
1
2
3
4
5
6


Boxplot de Garagem para Casa em Condomínio
Contagem
0
1
2
3
4
5
6


Boxplot de Aluguel para Casa em Condomínio
Contagem
0
2000
4000
6000
8000
10000
12000
14000



Pelos gráficos já podemos entender um pouco melhor do nosso conjunto de dados.

```
medias_por_tipo = df_bruto.groupby('Tipo').mean(numeric_only=True)

medias_por_tipo.head()
```

	Area	Quartos	Garagem	Aluguel	IPTU
Tipo					
Apartamento	73.318460	1.987907	1.022519	3356.902697	1078.525716
Casa	136.136220	2.353749	1.514960	3471.924674	352.319606
Casa em condomínio	119.414938	2.236515	1.414938	3912.551867	663.680498
Studio e kitnet	31.742216	1.009413	0.260681	2127.825489	540.454743

Comparando os extremos, podemos observar:

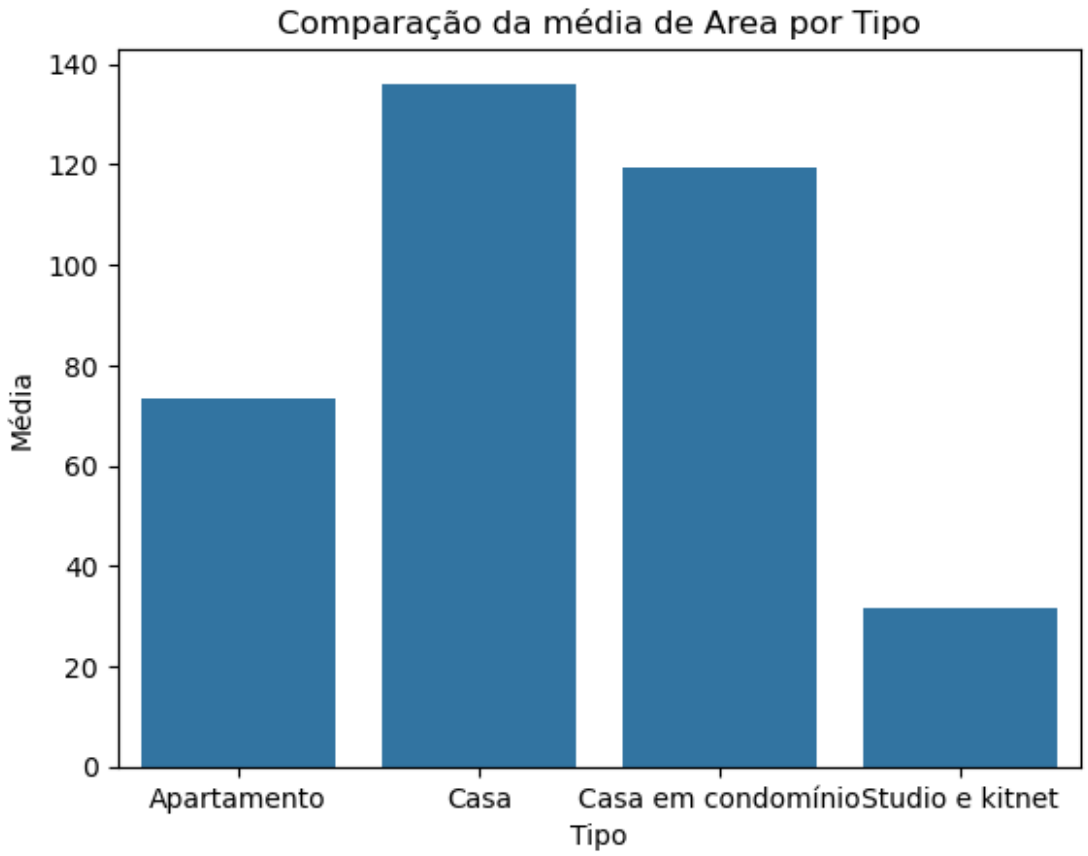
1 - 'Studio e kitnet' é o que possui a menor média em área, quartos, garagem e aluguel.

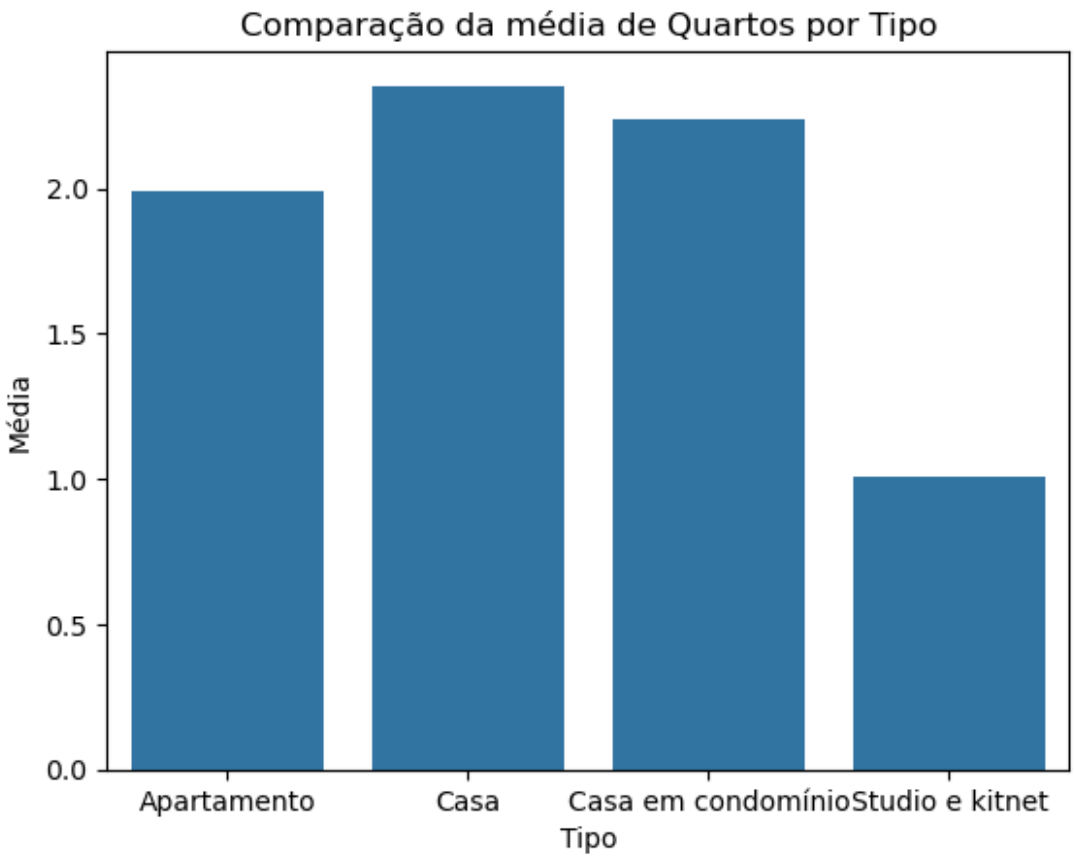
2 - 'Casa' é o que possui a maior média em área, quartos e garagem.

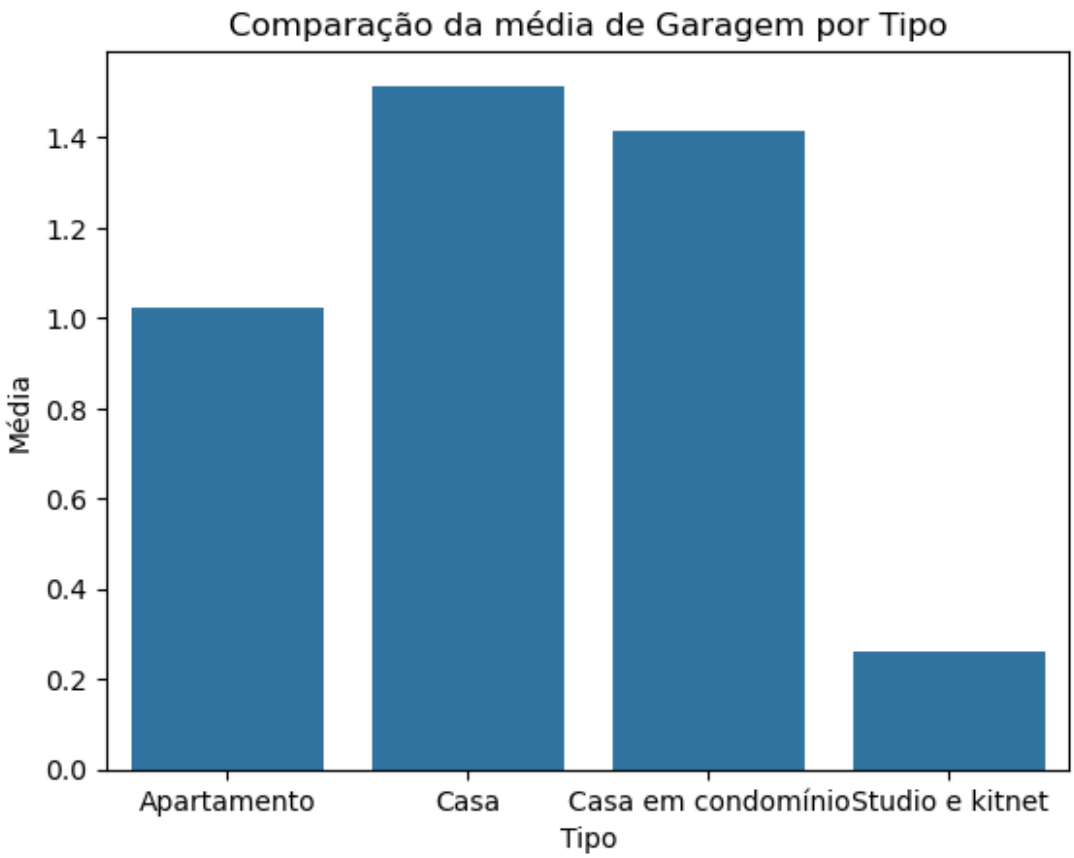
3 - 'Casa em condomínio' é o que possui a maior média em aluguel.

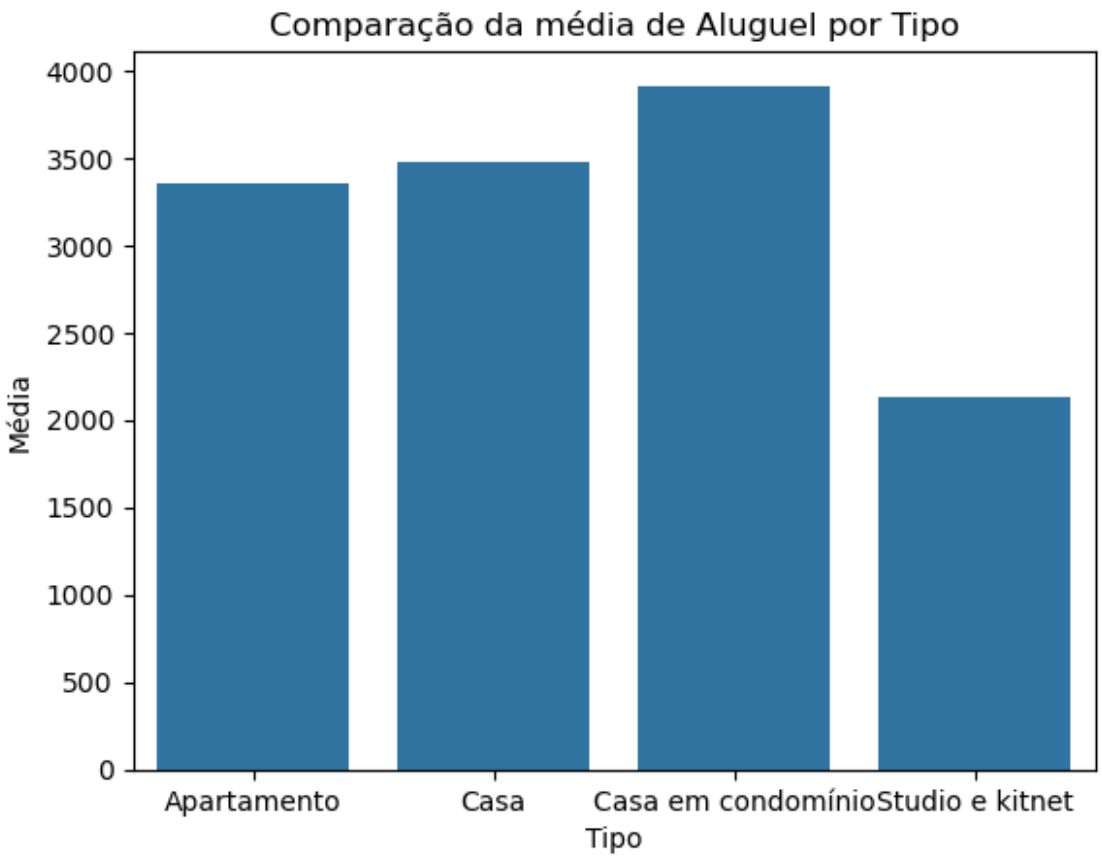
4 - 'Apartamente' é o que possui a maior média em iptu.

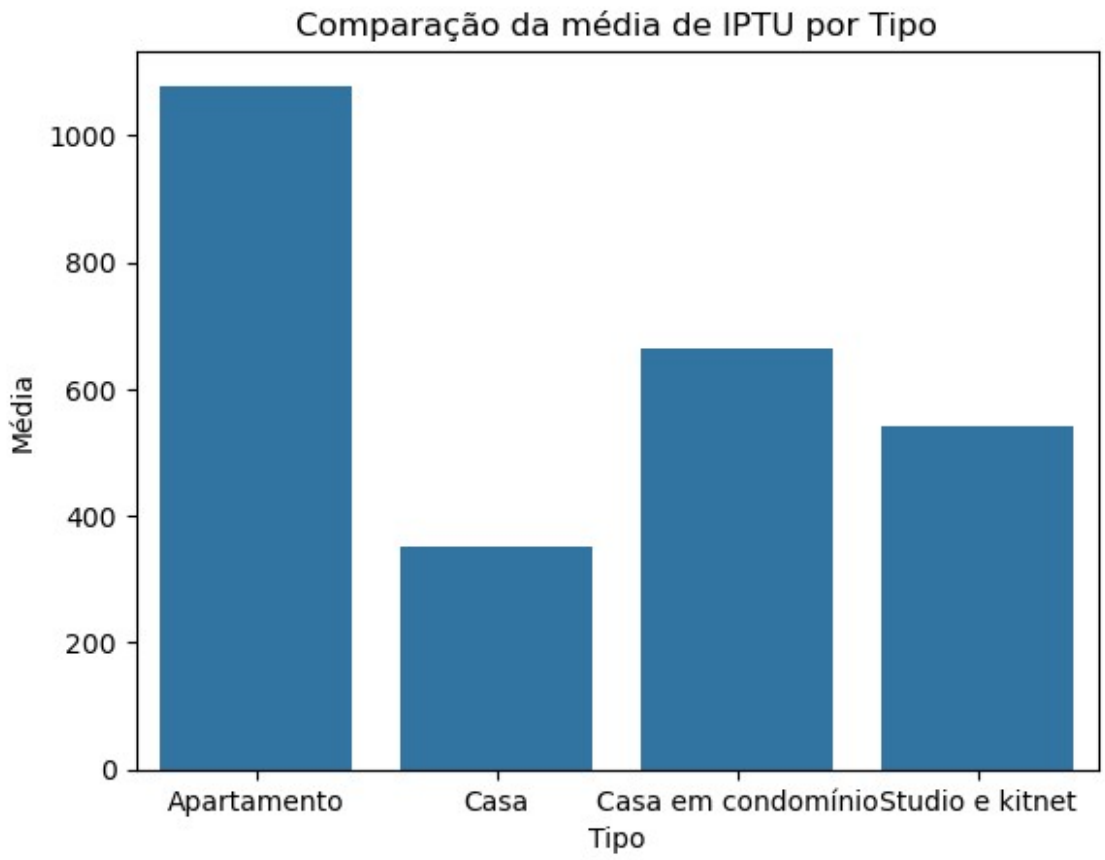
```
for column in colunas_quantitativas:
    sns.barplot(medias_por_tipo[column])
    plt.title(f'Comparação da média de {column} por Tipo')
    plt.ylabel('Média')
    plt.show()
```




Comparação da média de Quartos por Tipo
Média
2.0
1.5
1.0
0.5
0.0
Apartamento
Casa
Casa em condomínio
Studio e kitnet
Tipo


Comparação da média de Garagem por Tipo
Média
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Apartamento
Casa
Casa em condomínio
Studio e kitnet
Tipo


Comparação da média de Aluguel por Tipo
Média
4000
3500
3000
2500
2000
1500
1000
500
0
Apartamento
Casa
Casa em condomínio
Studio e kitnet
Tipo



3. Tratamento dos Dados

Agora vamos começar a realizar alguns ajustes no conjunto de dados, ainda buscando entender melhor o conjunto de dados e corrigir possíveis erros.

Após verificarmos que no conjunto de dados não há dados nulos ou vazios, decidimos realizar uma busca a partir de resultados lógicos.

A primeira busca é por imóveis em que sua área e o seu número de quartos sejam igual a zero.

Para a relação da área, sua justificativa é direta, não faz sentido um imóvel sem área.

Já para o número de quartos, é possível pensar que talvez para o tipo de imóvel "Studio e kitnet" esse valor possa ser justificado de alguma forma, então temos analisar com cuidado esse tipo de imóvel.

```
df_bruto_quartos_zero = df_bruto.query('Quartos == 0 | Area == 0')
print(df_bruto_quartos_zero.shape)
df_bruto_quartos_zero
```

```
(39, 8)
```

```
                                Endereço
Distrito  Area  \
71               Rua Natividade Saldanha                      São
Lucas    22
98               Rua Natividade Saldanha                      São
Lucas    19
163                Rua João José Pacheco            Jardim Vila
Mariana    22
259                  Rua Riskallah Jorge  Centro Histórico de São
Paulo    28
476    Rua Doutor Miguel Vieira Ferreira
Tatuapé    30
493              Rua Natividade Saldanha                      São
Lucas    32
569           Rua Albino Boldasso Gabril                   Vila
Cruzeiro    44
604              Rua Natividade Saldanha                      São
Lucas    29
631                     Avenida São João                    Santa
Cecilia    35
708           Rua Doutor Albuquerque Lins                    Santa
Cecília    63
1002             Rua Natividade Saldanha                      São
Lucas    28
1110                  Rua Padre Machado                 Bosque da
Saúde    35
1180                     Rua Nova Barão
República    44
1416             Rua Natividade Saldanha                      São
Lucas    27
1538             Rua Natividade Saldanha                      São
Lucas    38
1552             Rua Natividade Saldanha                      São
Lucas    40
1595             Rua Natividade Saldanha                      São
Lucas    45
1596                      Rua Marilândia                  Freguesia do
Ó    70
1925                    Rua Santa Lúcia              Cidade Mãe do
Céu    22
1965             Rua Natividade Saldanha                      São
Lucas    25
2177                    Rua Bela Cintra
Consolação    25
2983                       Rua Relíquia         Jardim das
Laranjeiras     0
3059                    Rua Frei Caneca
Consolação    26
4729                 Rua Joaquim Guarani             Jardim das
Acacias    19
```

```
5864                  Rua Azevedo Marques                    Santa
Cecília    34
6493                          Rua Dráusio
Butantã    28
7085                Rua Edmundo de Amicis
Morumbi     0
7136                      Rua Manoel Dutra                      Bela
Vista    22
7366                Rua Doutor Ângelo Vita                      Vila
Zilda    31
7587                      Largo do Arouche
República    30
8240                    Rua Santa Madalena
Liberdade    20
8308                    Rua dos Carmelitas
Sé    38
8343     Avenida Brigadeiro Luís Antônio                        Bela
Vista    40
8422                Rua Dom José de Baros
Centro    40
9265                Rua Azem Abdalla Azem              Jardim
Bonfiglioli    14
9501             Rua Dom Armando Lombardi                 Vila
Progredior   250
9674                  Rua Adalberto Kemeny  Parque Industrial Tomas
Edson    47
10062                           Rua Graúna                 Vila
Uberabinha   141
10082           Rua Conde de Porto Alegre                       Campo
Belo   200

       Quartos  Garagem             Tipo  Aluguel  IPTU
71           0        0  Studio e kitnet      900   226
98           0        0  Studio e kitnet      850   202
163          0        0  Studio e kitnet     2200   608
259          0        0  Studio e kitnet     1207   440
476          0        0      Apartamento     1840   273
493          0        0  Studio e kitnet     1200   332
569          0        0  Studio e kitnet     1400   198
604          0        0  Studio e kitnet     1150   302
631          0        0  Studio e kitnet      935   413
708          0        0  Studio e kitnet     1600   586
1002         0        0      Apartamento     1100   291
1110         0        0  Studio e kitnet     1750    68
1180         0        0  Studio e kitnet     1550   610
1416         0        0      Apartamento     1100   287
1538         0        0  Studio e kitnet     1300   388
1552         0        0      Apartamento     1300   406
1595         0        0      Apartamento     1400   455
1596         0        0             Casa     3100    52
```

```
1925          0         0      Apartamento     1000   263
1965          0         0  Studio e kitnet     1050   264
2177          0         0  Studio e kitnet     3500   835
2983          2         1      Apartamento     1600   969
3059          0         0  Studio e kitnet     2810   496
4729          0         0  Studio e kitnet     1970   434
5864          0         0  Studio e kitnet     1139   424
6493          0         0  Studio e kitnet     2600   464
7085          3         2             Casa     3200   151
7136          0         0  Studio e kitnet     1700   206
7366          0         0  Studio e kitnet     1700   666
7587          0         0  Studio e kitnet     1010   458
8240          0         0  Studio e kitnet      850   183
8308          0         0  Studio e kitnet      980   504
8343          0         1  Studio e kitnet     2700   856
8422          0         0  Studio e kitnet      990   389
9265          0         0  Studio e kitnet     1050    48
9501          0         3             Casa    10000   290
9674          0         1      Apartamento     2330   860
10062         0         0             Casa     7000   208
10082         0         4             Casa    12500   290
```

A nossa query apresentou 39 imóveis com "Quartos == 0 | Area == 0", comparando com o tamanho do nosso conjunto de dados, é um número pequeno.

Por ser uma quantidade baixa, poderiamos verificar realizar alguma ação de forma individual, linhas por linha.

Porém, vamos analisar por meio de quantos % esses imóveis representam em cada tipo.

```
df_bruto_quartos_zero_values = df_bruto_quartos_zero.reset_index()
values_quartos_zero =
pd.DataFrame(df_bruto_quartos_zero_values['Tipo'].value_counts()).rese
t_index()
values_quartos_zero.columns = ['Tipo', 'Quantidades']
values_quartos_zero
```

```
              Tipo  Quantidades
0  Studio e kitnet           26
1      Apartamento            8
2             Casa            5
```

```
values_bruto_tipo =
pd.DataFrame(df_bruto['Tipo'].value_counts()).reset_index()
values_bruto_tipo.columns = ['Tipo', 'Quantidades']
values_bruto_tipo
```

```
                Tipo  Quantidades
0         Apartamento         7194
1                Casa         2841
```

```
2     Studio e kitnet         1381
3  Casa em condomínio          241
```

```
studiokitnet_1 = values_quartos_zero.loc[0, 'Quantidades']
apartamento_1 = values_quartos_zero.loc[1, 'Quantidades']
casa_1 = values_quartos_zero.loc[2, 'Quantidades']

studiokitnet_2 = values_bruto_tipo.loc[2, 'Quantidades']
apartamento_2 = values_bruto_tipo.loc[0, 'Quantidades']
casa_2 = values_bruto_tipo.loc[1, 'Quantidades']

porcentagem_studiokitnet = (studiokitnet_1 / studiokitnet_2)*100
print(f'A quantidade de Studio/Kitnet com "Quartos == 0 | Area ==0"
representa {porcentagem_studiokitnet.round(2)}% do conjunto total.')
```

```
A quantidade de Studio/Kitnet com "Quartos == 0 | Area ==0" representa
1.88% do conjunto total.
```

```
porcentagem_apartamento = (apartamento_1 / apartamento_2)*100
print(f'A quantidade de Apartamentos com "Quartos == 0 | Area ==0"
representa {porcentagem_apartamento.round(2)}% do conjunto total.')
```

```
A quantidade de Apartamentos com "Quartos == 0 | Area ==0" representa
0.11% do conjunto total.
```

```
porcentagem_casa = (casa_1 / casa_2)*100
print(f'A quantidade de Casa com "Quartos == 0 | Area ==0" representa
{porcentagem_casa.round(2)}% do conjunto total.')
```

```
A quantidade de Casa com "Quartos == 0 | Area ==0" representa 0.18% do
conjunto total.
```

Como esses valores de fato representam uma porcentagem muito pequena do conjunto total, vamos apenas removê-las.

```
registros_a_remover = df_bruto_quartos_zero.index

df_filtrado = df_bruto.drop(registros_a_remover, axis=0)

df_filtrado.query('Area == 0 | Quartos == 0')
```

```
Empty DataFrame
Columns: [Endereço, Distrito, Area, Quartos, Garagem, Tipo, Aluguel,
IPTU]
Index: []
```

Pela proposta do projeto, um os objetivos que devemos realizar é ajustar um modelo de regressão linear aos dados para tentar prever o preço do aluguel em uma determinada área.

Vamos interpretar que "determinada área" diz respeito sobre região geográfica da cidade.

Como temos duas categorias que nos informam sobre isso, temos que analisa-lás.

```
df_filtrado['Endereço'].value_counts()
```

```
Endereço
Rua da Consolação                     49
Rua Bela Cintra                       46
Avenida Brigadeiro Luís Antônio       35
Avenida Ipiranga                      32
Avenida Nove de Julho                 29
                                      ..
Dona Maria Pera                        1
Rua Passo da Pátria                    1
Rua Teixeira Leite                     1
Avenida Professor Abraão de Morais     1
Rua Abílio Borin                       1
Name: count, Length: 5345, dtype: int64
```

```
df_filtrado['Distrito'].value_counts()
```

```
Distrito
Bela Vista               350
Vila Mariana             232
Jardim Paulista          220
Centro                   177
Pinheiros                159
                        ...
Jardim do Carmo            1
Santa Inês                 1
Jardim Santa Efigenia      1
Vila Maricy                1
Retiro Morumbi             1
Name: count, Length: 1199, dtype: int64
```

Pela nossa busca, identificamos que a categoria que melhor agrupa o conjunto de dados é a categoria 'Distrito'.

Então vamos remover a categoria 'Endereço'.

```
df_filtrado.drop('Endereço', axis=1, inplace=True)
```

```
df_filtrado.head()
```

```
       Distrito  Area  Quartos  Garagem                Tipo  Aluguel
IPTU
0    Belenzinho    21        1        0     Studio e kitnet     2400
539
1  Vila Marieta    15        1        1     Studio e kitnet     1030
315
2     Pinheiros    18        1        0         Apartamento     4000
661
3       Vila Ré    56        2        2  Casa em condomínio     1750
```

```
204
4    Bela Vista    19        1        0     Studio e kitnet     4000
654
```

Surge uma nova questão, temos 1199 distritos diferentes no conjunto de dados, sendo que a lista oficial de distritos para a cidade de São Paulo possui apenas 96.

A partir disso, temos duas hipoteses que podem explicar essa discrepância:

1° - O conjunto de dados é sobre a região metropolitana de São Paulo, que reúne 39 municípios.

2° - O conjunto tem variação na escrita do nome dos distritos, então o mesmo distrito pode estar sendo contado individualmente apenas por ter algum variação na forma que seu nome foi escrito.

4. Preparação dos Dados

Por causa da situação encontrada na categoria 'Distrito', vamos testar algumas manipulações buscando tornar o conjunto de dados mais organizado.

Nesse primeiro momento, vamos preparar 2 conjuntos de dados para testar a modelagem por regressão linear:

Modelo 1 - Conjunto de dados apenas com os distritos oficiais.

Modelo 2 - Conjunto de dados dos distritos oficiais e agrupados pelas regiões (central, norte, leste, oeste e sul).

Antes de mais nada, vamos conferir como estão as correlações do conjunto de dados.

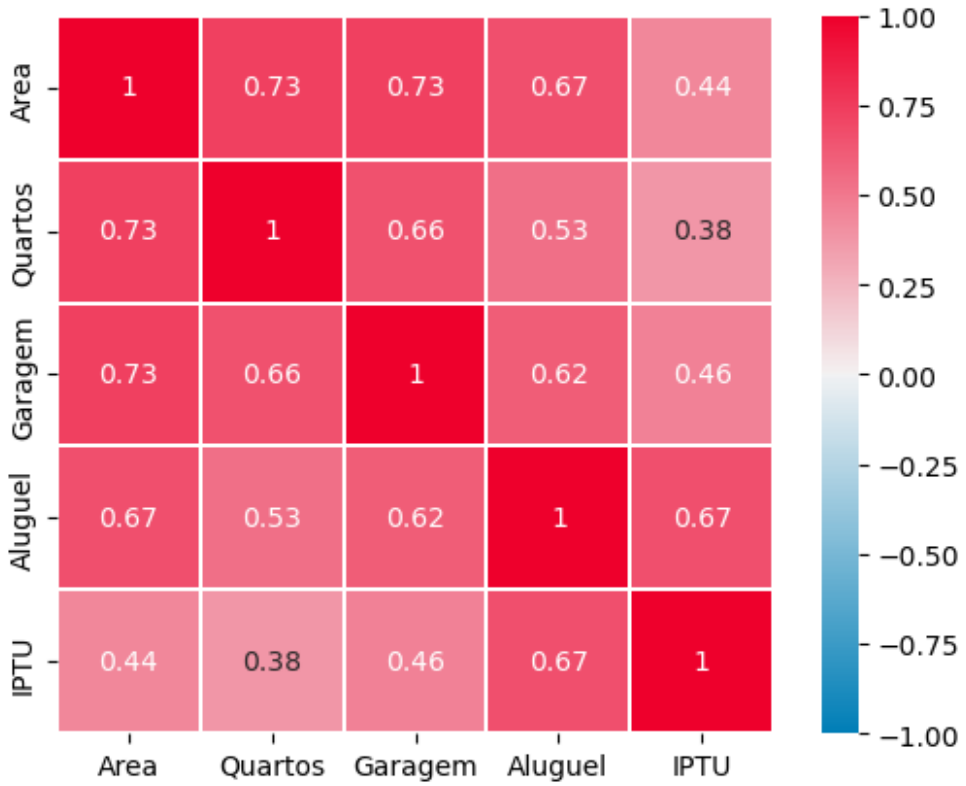
```
df_corr = df_filtrado[colunas_quantitativas]
df_corr.corr()
```

	Area	Quartos	Garagem	Aluguel	IPTU
Area	1.000000	0.730630	0.733156	0.666598	0.444892
Quartos	0.730630	1.000000	0.657800	0.533265	0.378249
Garagem	0.733156	0.657800	1.000000	0.616563	0.463822
Aluguel	0.666598	0.533265	0.616563	1.000000	0.670083
IPTU	0.444892	0.378249	0.463822	0.670083	1.000000

```
cmap = sns.diverging_palette(
    h_neg=240,
    h_pos=10,
    s=100,
    as_cmap=True,
)
sns.heatmap(
    df_corr.corr(),
    cmap=cmap,
```

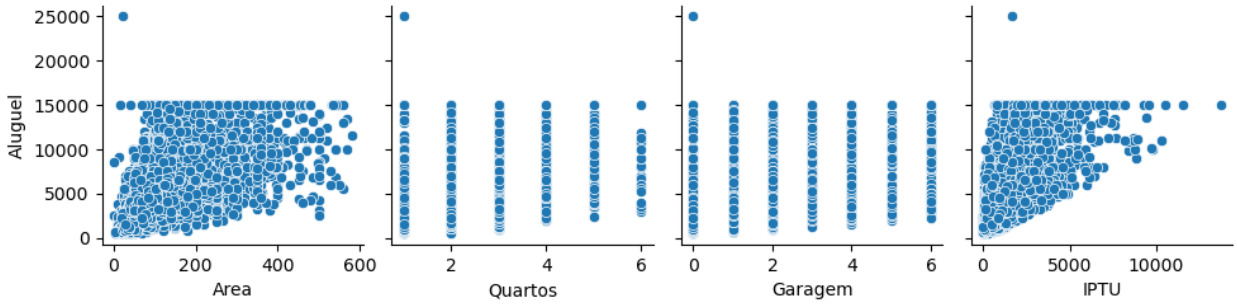
```
    center=0,
    vmin=-1,
    vmax=1,
    square=True,
    linewidths=0.01,
    annot=True,
    xticklabels=colunas_quantitativas,
    yticklabels=colunas_quantitativas,
)
```

```
<Axes: >
```




```
sns.pairplot(df_filtrado, y_vars='Aluguel', x_vars=['Area', 'Quartos',
'Garagem', 'IPTU'])
```

```
<seaborn.axisgrid.PairGrid at 0x270487d0d60>
```



Agora vamos adicionar índices as categorias 'Tipo' e 'Distrito', para que posssamos transforma-lás em variáveis dummy para o modelo.

```
df_prep1 = df_filtrado.copy()

tipos_imoveis = df_prep1['Tipo'].unique()
tipos_id = {tipo: idx for idx, tipo in enumerate(tipos_imoveis,
start=1)}
print(tipos_id)
```

```
{'Studio e kitnet': 1, 'Apartamento': 2, 'Casa em condomínio': 3,
'Casa': 4}
```

```
df_prep1['Tipo_ID'] = df_prep1['Tipo'].map(tipos_id)

df_prep1.head()
```

```
       Distrito  Area  Quartos  Garagem                Tipo  Aluguel
IPTU  \
0    Belenzinho    21        1        0     Studio e kitnet     2400
539
1  Vila Marieta    15        1        1     Studio e kitnet     1030
315
2     Pinheiros    18        1        0         Apartamento     4000
661
3       Vila Ré    56        2        2  Casa em condomínio     1750
204
4    Bela Vista    19        1        0     Studio e kitnet     4000
654

   Tipo_ID
0        1
1        1
2        2
3        3
4        1
```

```
distritos_unicos = df_prep1['Distrito'].unique()
distrito_id = {distrito: idx for idx, distrito in
enumerate(distritos_unicos, start=1)}
print(distrito_id)
```

{'Belenzinho': 1, 'Vila Marieta': 2, 'Pinheiros': 3, 'Vila Ré': 4, 'Bela Vista': 5, 'Brás': 6, 'Brooklin Paulista': 7, 'Centro': 8, 'Piqueri': 9, 'Vila Aricanduva': 10, 'Sé': 11, 'Tatuapé': 12, 'Lauzane Paulista': 13, 'Jardim Paraventi': 14, 'Cambuci': 15, 'Liberdade': 16, 'Cidade Monções': 17, 'Água Branca': 18, 'Mooca': 19, 'Chácara Inglesa': 20, 'Vila Moreira': 21, 'Gopoúva': 22, 'Jardim São Savério': 23, 'Vila Amalia (zona Norte)': 24, 'Saúde': 25, 'Quarta Parada': 26, 'Santa Efigênia': 27, 'Paraíso do Morumbi': 28, 'Chora Menino': 29, 'Vila Medeiros': 30, 'Vila Guarani(zona Leste)': 31, 'Barra Funda': 32, 'Vila Augusta': 33, 'Vila Olímpia': 34, 'Vila Celeste': 35, 'Jardim Independência': 36, 'Vila Buarque': 37, 'Vila Vivaldi': 38, 'Vila Brasílio Machado': 39, 'Vila Carlos de Campos': 40, 'Vila Prudente': 41, 'Vila Andrade': 42, 'Campos Elíseos': 43, 'Vila Nivi': 44, 'Vila Palmeiras': 45, 'Ponte Grande': 46, 'Vila Marina': 47, 'Jardim Pereira Leite': 48, 'Jardim Utinga': 49, 'República': 50, 'Vila São Luís(zona Oeste)': 51, 'Vila Guilherme': 52, 'Santa Ifigênia': 53, 'Vila Galvão': 54, 'Vila Monumento': 55, 'Vila Penteado': 56, 'Vila Alpina': 57, 'Picanço': 58, 'Sacomã': 59, 'Vila Moinho Velho': 60, 'Vila Santa Maria': 61, 'Várzea da Barra Funda': 62, 'Vila Talarico': 63, 'Maranhão': 64, 'Vila Caminho do Mar': 65, 'Vila Santa Clara': 66, 'Macedo': 67, 'Bosque da Saúde': 68, 'Vila Constança': 69, 'Parque Fongaro': 70, 'Jardim das Acácias': 71, 'Penha de França': 72, 'Vila São Paulo': 73, 'Vila Sofia': 74, 'Campo Grande': 75, 'Cerqueira César': 76, 'Santa Teresinha': 77, 'Alto da Lapa': 78, 'Sítio do Morro': 79, 'Vila Ipojuca': 80, 'Jaguaré': 81, 'Parada Inglesa': 82, 'Vila Sao Pedro': 83, 'Vila Milton': 84, 'Jardim Flor da Montanha': 85, 'Ipiranga': 86, 'Consolação': 87, 'Vila Campestre': 88, 'Luz': 89, 'Jardim Alvorada (zona Oeste)': 90, 'Vila Arcadia': 91, 'Vila Mangalot': 92, 'Tremembé': 93, 'Vila Suzana': 94, 'Jardim Peri': 95, 'Tucuruvi': 96, 'Cidade Patriarca': 97, 'Campo Belo': 98, 'Engenheiro Goulart': 99, 'Vila Pedra Branca': 100, 'Santa Cecília': 101, 'Vila Mariana': 102, 'Centro Histórico de São Paulo': 103, 'Vila Indiana': 104, 'Vila Vermelha': 105, 'Jardim Maringá': 106, 'Jardim Bonfiglioli': 107, 'Vila Butantã': 108, 'Vila Gumercindo': 109, 'Sítio da Figueira': 110, 'Sumarezinho': 111, 'Santana': 112, 'Vila da Saúde': 113, 'Vila Mazzei': 114, 'Jardim Leonor': 115, 'Santo Amaro': 116, 'Jardim Vila Galvão': 117, 'Jardim Iris': 118, 'Campestre': 119, 'Vila Pirajussara': 120, 'Jardim Planalto': 121, 'Vila Moraes': 122, 'Vila Endres': 123, 'Vila Monte Alegre': 124, 'Chácara Belenzinho': 125, 'Jardim Sao Saverio': 126, 'Planalto Paulista': 127, 'Real Parque': 128, 'Pacaembu': 129, 'Paulicéia': 130, 'Taboão': 131, 'Jardim Mariliza': 132, 'Jardim Ampliação': 133, 'Vila Rosália': 134, 'Jardim São José (zona Norte)': 135, 'Jardim Clímax': 136, 'Vila Curuca': 137, 'Jardim Olympia': 138, 'Vila Mendes': 139, 'Vila Guarani (z Sul)': 140, 'Casa Verde': 141, 'Vila Mascote': 142, 'Vila Maria Alta': 143, 'Vila Leopoldina': 144, 'Jardim Santa Emilia': 145, 'Vila Santa Catarina': 146, 'Vila Santo Estéfano': 147, 'Limão': 148, 'Perdizes': 149, 'Vila Clementino': 150, 'Jardim Piqueroby': 151, 'Jardim Paulista': 152, 'Vila Gomes Cardim': 153, 'Vila Cruzeiro': 154, 'Jardim Anália Franco': 155, 'Jardim Aeroporto': 156, 'Rudge

Ramos': 157, 'Sacoma': 158, 'Vila Esperança': 159, 'Quinta da Paineira': 160, 'Vila Amália (zona Norte)': 161, 'Jardim Hercilia': 162, 'Vila Santa Luzia': 163, 'Paraíso': 164, 'Vila Firmiano Pinto': 165, 'Vila Cordeiro': 166, 'Vila Dom Pedro Ii': 167, 'Vila Siqueira (zona Norte)': 168, 'Alto da Mooca': 169, 'Vila Madalena': 170, 'Indianópolis': 171, 'Vila Ema': 172, 'Vila Anglo Brasileira': 173, 'Jardim Santa Maria': 174, 'Santa Paula': 175, 'Americanópolis': 176, 'Vila Bela': 177, 'Vila Albertina': 178, 'Chácara Mafalda': 179, 'Vila Nova Savoia': 180, 'Brooklin': 181, 'Vila Valparaíso': 182, 'Jurubatuba': 183, 'Casa Verde Alta': 184, 'Vila Matilde': 185, 'Jardim Las Vegas': 186, 'Jardim Peri Peri': 187, 'Cidade Mãe do Céu': 188, 'Santa Cecilia': 189, 'Vila Venditti': 190, 'Vila Inah': 191, 'Lapa': 192, 'Jardim América da Penha': 193, 'Vila Nova Carolina': 194, 'Parque Novo Mundo': 195, 'Jardim da Glória': 196, 'Vila Guedes': 197, 'Chácara Santo Antônio (zona Sul)': 198, 'Vila Sonia': 199, 'Jardim Celeste': 200, 'Vila Regina': 201, 'Cangaíba': 202, 'Socorro': 203, 'Jardim Rosa de Franca': 204, 'Catumbi': 205, 'Vila Pereira Cerca': 206, 'Lajeado': 207, 'Vila Pompéia': 208, 'Vila Isa': 209, 'Jardim Oriental': 210, 'Mirandópolis': 211, 'Jardim Paulistano': 212, 'Vila Margarida': 213, 'Vila Nova Mazzei': 214, 'Vila Ede': 215, 'Sapopemba': 216, 'Butantã': 217, 'Jabaquara': 218, 'Nossa Senhora do Ó': 219, 'São Judas': 220, 'Jardim Prudência': 221, 'Vila Congonhas': 222, 'Água Fria': 223, 'Jardim Sao Francisco (zona Leste)': 224, 'Guaiaúna': 225, 'Vila Oratorio': 226, 'Vila Buenos Aires': 227, 'Vila Isolina Mazzei': 228, 'Parque Peruche': 229, 'Vila Cardoso Franco': 230, 'Super Quadra Morumbi': 231, 'Vila Romana': 232, 'Jardim Aurelia': 233, 'Vila Fidalgo': 234, 'Parque Industrial Tomas Edson': 235, 'Vila Formosa': 236, 'Jardim Aricanduva': 237, 'Vila Domitila': 238, 'Vila Água Funda': 239, 'Vila Beatriz': 240, 'Vila Paulicéia': 241, 'Imirim': 242, 'Parque Edu Chaves': 243, 'Vila Inglesa': 244, 'São Lucas': 245, 'Sítio do Mandaqui': 246, 'Cangaiba': 247, 'Jardim Rosana': 248, 'Jardim Sao Paulo(zona Norte)': 249, 'Aclimação': 250, 'Jardim Umarizal': 251, 'Vila Príncipe de Gales': 252, 'Vila Guarani': 253, 'Jardim das Vertentes': 254, 'Parque Tomas Saraiva': 255, 'Parque Vitória': 256, 'Jardim Pinhal': 257, 'Vila Graciosa': 258, 'Vila Canero': 259, 'Jardim Pazini': 260, 'Vila Invernada': 261, 'Jardim Santa Emília': 262, 'Vila Carrao': 263, 'Vila Cruz das Almas': 264, 'Jardim Brasília (zona Norte)': 265, 'Morumbi': 266, 'Vila Regente Feijó': 267, 'Parque Mandaqui': 268, 'Vila Sônia': 269, 'Jardim Tijuco': 270, 'Vila Vera': 271, 'Silveira': 272, 'Vila Antonieta': 273, 'Cidade Vargas': 274, 'Vila Rui Barbosa': 275, 'Jardim Lourdes (zona Sul)': 276, 'Vila Guilhermina': 277, 'Parque Jabaquara': 278, 'Moema': 279, 'Jardim Íris': 280, 'Jardim Adutora': 281, 'Vila Amélia': 282, 'Vila Barbosa': 283, 'Vila Ester (zona Norte)': 284, 'Vila Bancária': 285, 'Vila Rio de Janeiro': 286, 'Loteamento City Jaragua': 287, 'Jardim Parque Morumbi': 288, 'Vila Floresta': 289, 'Alto de Pinheiros': 290, 'Jardim Almanara': 291, 'Jardim Barbosa': 292, 'Vila Santa Teresa (zona Sul)': 293, 'Jardim Jabaquara': 294, 'Jardim Centenario': 295, 'Vila Aurora (zona Norte)': 296, 'Vila Nova Cachoeirinha': 297, 'Vila Santo Estefano': 298, 'Bom Retiro': 299,

'Vila Maria': 300, 'Utinga': 301, 'Jardim Esmeralda': 302, 'Vila Bertioga': 303, 'Parque Assunção': 304, 'Vila Nova Conceição': 305, 'Vila Miriam': 306, 'Vila Guiomar': 307, 'Jardim Miriam': 308, 'Jardim das Acacias': 309, 'Jardim Jaqueline': 310, 'Vila Zilda': 311, 'Parque Continental Ii': 312, 'Jardim Vila Mariana': 313, 'Vila Pirituba': 314, 'Vila Lageado': 315, 'Vila Ivone': 316, 'Higienópolis': 317, 'Vila Dom Pedro I': 318, 'Vila São Silvestre': 319, 'Parque Imperial': 320, 'Vila Gertrudes': 321, 'Jardim Caravelas': 322, 'Jardim Previdência': 323, 'Parque da Mooca': 324, 'Santa Maria': 325, 'Vila Santana': 326, 'Jardim Umuarama': 327, 'Portal dos Gramados': 328, 'Cidade Luz': 329, 'Vila Alexandria': 330, 'São João Climaco': 331, 'Vila Pires': 332, 'Itaim Bibi': 333, 'Jardim Colorado': 334, 'Vila Diva (zona Norte)': 335, 'Vila Nova Caledônia': 336, 'Cidade São Francisco': 337, 'Jardim Avelino': 338, 'Jardim Ampliacao': 339, 'Jardim Adriana': 340, 'Jardim Regina': 341, 'Jardim Mangalot': 342, 'Vila Silvia': 343, 'Vila Metalurgica': 344, 'Vila Jaguara': 345, 'Vila Santa Teresa': 346, 'Vila Baruel': 347, 'Vila Nossa Senhora de Fátima': 348, 'Jardim Gracinda': 349, 'Boa Vista': 350, 'Jardim Vergueiro (sacoma)': 351, 'Chácara São João': 352, 'Chácara Califórnia': 353, 'Vila Oratório': 354, 'Vila Gilda': 355, 'Vila Palmares': 356, 'Vila Nova York': 357, 'Jardim Paraíso': 358, 'Rio Pequeno': 359, 'Campanário': 360, 'Vila Diva (zona Leste)': 361, 'Vila Guaca': 362, 'Vila Maracanã': 363, 'Carandiru': 364, 'Vila São Luís': 365, 'Vila Rio': 366, 'Mandaqui': 367, 'Jardim Jamaica': 368, 'Vila Valparaiso': 369, 'Parque Casa de Pedra': 370, 'Vila Guarani (zona Sul)': 371, 'Vila Pierina': 372, 'Vila Nair': 373, 'Alto do Pari': 374, 'Vila Natália': 375, 'Vila Leonor': 376, 'Vila Bremen': 377, 'Vila Sao Geraldo': 378, 'Vila Carrão': 379, 'Cursino': 380, 'Jardim Castelo': 381, 'Jardim Catanduva': 382, 'Jardim Brasil (zona Norte)': 383, 'Vila Francisco Matarazzo': 384, 'Vila das Merces': 385, 'Vila Salete': 386, 'Vila Divina Pastora': 387, 'Vila Sacadura Cabral': 388, 'Vila Boacava': 389, 'Jardim Santa Cruz (sacomã)': 390, 'Vila Fachini': 391, 'Vila Independencia': 392, 'Cidade Continental': 393, 'Jardim Casablanca': 394, 'Santo Antônio': 395, 'Vila Lais': 396, 'Vila do Castelo': 397, 'Olímpico': 398, 'Vila Pereira Barreto': 399, 'Sumaré': 400, 'Jardim Raposo Tavares': 401, 'Jardim Modelo': 402, 'Vila Portugal': 403, 'Vila Gomes': 404, 'Jd Tranquilidade ': 405, 'Vila Constancia': 406, 'Jardim do Tiro': 407, 'Jardim Angela (zona Leste)': 408, 'Vila Helena': 409, 'Parque Reboucas': 410, 'Parque das Nações': 411, 'Vila Santa Terezinha': 412, 'Vila Principe de Gales': 413, 'Jardim Europa': 414, 'Vila Maria Baixa': 415, 'Vila Caraguatá': 416, 'Parque Continental': 417, 'Vila Caraguata': 418, 'Fazenda Morumbi': 419, 'Cidade Domitila': 420, 'Vila Morumbi': 421, 'Jardim Caboré': 422, 'Chácara Santo Antônio (Zona Sul)': 423, 'Jardim Nosso Lar': 424, 'Vila Deodoro': 425, 'Vila Babilônia': 426, 'Jardim Vila Formosa': 427, 'Vila Parque Jabaquara': 428, 'Vila Uberabinha': 429, 'Jardim Vitória Régia': 430, 'Jardim das Maravilhas': 431, 'Jardim Vergueiro (sacomã)': 432, 'Jardim Leonor Mendes de Barros': 433, 'Jardim Brasil (zona Sul)': 434, 'Jardim do Mar': 435, 'Freguesia do Ó': 436, 'Vila Granada': 437, 'Vila Antonina': 438, 'Parque João

Ramalho': 439, 'Canindé': 440, 'Jardim Iporanga': 441, 'Vila Cristália': 442, 'Jardim Santa Mônica': 443, 'Mauá': 444, 'Chácara Santo Antônio (zona Leste)': 445, 'Vila Gustavo': 446, 'Vila Alto de Santo André': 447, 'Vila do Encontro': 448, 'Osvaldo Cruz': 449, 'Jardim São Paulo': 450, 'Jardim Ataliba Leonel': 451, 'Vila Sabrina': 452, 'Chácara Seis de Outubro': 453, 'Casa Verde Média': 454, 'Panamby': 455, 'Jardim Almeida Prado': 456, 'Vila Carbone': 457, 'Jardim Maria Estela': 458, 'Vila Clarice': 459, 'Jardim Botucatu': 460, 'Conjunto Residencial Vista Verde': 461, 'Jardim Pinheiros': 462, 'Vila Socorro': 463, 'Vila das Mercês': 464, 'Parque Bristol': 465, 'Jardim Seckler': 466, 'Vila das Belezas': 467, 'Jardim do Papai': 468, 'Jardim Tranqüilidade': 469, 'Vila Nova Manchester': 470, 'Itaberaba': 471, 'Vila São Pedro': 472, 'Jardim das Perdizes': 473, 'Vila Paulo Silas': 474, 'Jardim Popular': 475, 'Vila California': 476, 'Jardim Antartica': 477, 'Vila Progredior': 478, 'Vila Brasilina': 479, 'Caxingui': 480, 'Jardim Bela Vista': 481, 'Vila Marte': 482, 'Vila Cleonice': 483, 'Vila Mira': 484, 'República ': 485, 'Vila Renata': 486, 'Ponte Pequena': 487, 'Brás ': 488, 'Jardim Patente Novo': 489, 'Vila Anastácio': 490, 'Jardim Arize': 491, 'Parque da Lapa': 492, 'Jardim Sao Judas Tadeu': 493, 'Pompeia': 494, 'Vila Sao Francisco (zona Sul)': 495, 'Vila São Ricardo': 496, 'Brooklin Novo': 497, 'Vila Camilópolis': 498, 'Paraisópolis': 499, 'Jardim Marajoara': 500, 'Jardim das Laranjeiras': 501, 'Vila Santa Virginia': 502, 'Parque Monteiro Soares': 503, 'Jardim Divinolandia': 504, 'Jardim Vera Cruz': 505, 'Vila Planalto': 506, 'Jardim de Lorenzo': 507, 'Parque da Vila Prudente': 508, 'Nova Gerti': 509, 'Jardim Dona Sinhá': 510, 'Jaguará': 511, 'Vila Basileia': 512, 'Vila Arruda': 513, 'Jardim Independência (são Paulo)': 514, 'Jardim Cidade Pirituba': 515, 'Jardim Boa Vista (zona Oeste)': 516, 'Jardim Piracuama': 517, 'Vila Sao Luis(zona Oeste)': 518, 'Vila Industrial': 519, 'Vila Parapuã': 520, 'Jardim Santo Elias': 521, 'Vila Morse': 522, 'Jardim das Gracas': 523, 'Jardim Valéria': 524, 'Jardim Ângela (zona Leste)': 525, 'Jardim São Paulo(zona Norte)': 526, 'Jardim Santo Amaro': 527, 'São Domingos': 528, 'Jardim': 529, 'Canhema': 530, 'Jardim Toscana': 531, 'Vila Maria Trindade': 532, 'Vila Dalila': 533, 'Jardim Bom Clima': 534, 'Vila Sao Vicente': 535, 'Vila Romero': 536, 'Parque Sao Luis': 537, 'Jardim Campanario': 538, 'Vila Erna': 539, 'Jardim Taboão': 540, 'Republica': 541, 'Jardim Santa Edwiges (capela do Socorro)': 542, 'Vila Nova Alba': 543, 'Vila Bandeirantes': 544, 'Vila Marari': 545, 'Jardim Sílvia': 546, 'Jardim Guairaca': 547, 'Vila Santo Estevão': 548, 'Vila Paulista': 549, 'Vila Santa Isabel': 550, 'Vila Humaitá': 551, 'Instituto de Previdencia': 552, 'Fundação': 553, 'Água Rasa': 554, 'Jardim Alzira': 555, 'Jardim Santa Mena': 556, 'Vila União (zona Leste)': 557, 'Vila Iório': 558, 'Vila Alzira': 559, 'Sítio do Piqueri': 560, 'Vila São José (ipiranga)': 561, 'Anchieta': 562, 'Jardim Iva': 563, 'Vila Azevedo': 564, 'Jardim da Saúde': 565, 'Jardim Paramount': 566, 'Parque Fongaro ': 567, 'Jaraguá': 568, 'Vila Tolstoi': 569, 'Vila Souza': 570, 'Pari': 571, 'Jardim Marilu': 572, 'Vila Brasilio Machado': 573, 'Jaçanã': 574, 'Vila Nova': 575, 'Jardim Sao Joao (jaragua)': 576, 'Vila dos Andrades': 577, 'Vila Alianca':

578, 'Jardim Londrina': 579, 'Jardim Palmira': 580, 'Vila Scarpelli': 581, 'Nova Piraju': 582, 'Vila Babilonia': 583, 'Vila Assunção': 584, 'Jardim Previdencia': 585, 'Vila Arapuã': 586, 'Vila Plana': 587, 'Vila Paiva': 588, 'Jardim Antártica': 589, 'Jardim Nova Taboao': 590, 'Conjunto Residencial Jardim Canaã': 591, 'Vila Cachoeira': 592, 'Jardim Santa Clara': 593, 'Vila Bastos': 594, 'Jardim Gumercindo': 595, 'Jardim Aida': 596, 'Vila Califórnia': 597, 'Jardim Santa Terezinha': 598, 'Moinho Velho': 599, 'Jardim Jaú (Zona Leste)': 600, 'Jardim Lourdes': 601, 'Vila Algusta': 602, 'Vila Flórida': 603, 'Parque Oratorio': 604, 'Vila Eldízia': 605, 'Vila Natalia': 606, 'Vila Sao Paulo': 607, 'Jardim Julieta': 608, 'Continental': 609, 'Jardim Piratininga': 610, 'Jardim Centenário': 611, 'Instituto de Previdência': 612, 'Parque São Jorge': 613, 'Jardim Ibitirama': 614, 'Vila Bonilha Nova': 615, 'Vila Primavera': 616, 'Jardim Rizzo': 617, 'Parque Arariba': 618, 'Parque Maria Domitila': 619, 'City América': 620, 'Guapira': 621, 'Gopouva': 622, 'Vila Mesquita': 623, 'Vila Nova Santa Luzia': 624, 'Vila Hebe': 625, 'Jardim Canhema': 626, 'Jardim Novo Santo Amaro': 627, 'Jardim Danfer': 628, 'Jardim Cidália': 629, 'São José': 630, 'Vila N Sra de Fatima': 631, 'Jardim Consorcio': 632, 'Jardim Japão': 633, 'Brasilândia': 634, 'Jardim Líbano': 635, 'Vila Elze': 636, 'Jardim Vista Linda': 637, 'Parque Cruzeiro do Sul (vila Formosa)': 638, 'Vila Nilo': 639, 'Vila Progresso': 640, 'Jardim Trussardi': 641, 'Vila Camilopolis': 642, 'Vila Dinorah': 643, 'Chácara Monte Alegre': 644, 'Jardim Taquaral': 645, 'Jardim Guanca': 646, 'Alto da Boa Vista': 647, 'Vila Ramos': 648, 'Rolinópolis': 649, 'Vila Nancy': 650, 'Jardim Ivana': 651, 'Parque Vitoria': 652, 'Vila Roque': 653, 'Vila Silveira': 654, 'Vila Germinal': 655, 'Vila Paulistania': 656, 'Vila Tramontano': 657, 'Vila Continental': 658, 'Chácara do Vovó': 659, 'Vila Curuçá': 660, 'Vila Bertioga ': 661, 'Conjunto Habitacional Teotonio Vilela': 662, 'Vila Nova Pauliceia': 663, 'Vila Libanesa': 664, 'Jardim Cristin Alice': 665, 'Parque São Luís': 666, 'Ferreira': 667, 'Jardim Japao': 668, 'Jardim Testae': 669, 'Parque Alves de Lima': 670, 'Jardim Dom Bosco': 671, 'Jardim Dourado': 672, 'Parque Maria Luiza': 673, 'Jardim Montreal': 674, 'Vila Araguaia': 675, 'Vila Santa Edwiges': 676, 'Jardim São Francisco (zona Leste)': 677, 'Vila São José': 678, 'Jardim Cotiana': 679, 'Jardim Fonte do Morumbi': 680, 'Vila Euthalia': 681, 'Jardim Baruch': 682, 'Vila Serralheiro': 683, 'Vila Brasil': 684, 'Jardim Monte Kemel': 685, 'Vila São Domingos': 686, 'Vila Mariza Mazzei': 687, 'Vila das Bandeiras': 688, 'Vila Cláudia': 689, 'Vila Trabalhista': 690, 'Jardim Iracema': 691, 'Vila Bancaria': 692, 'Vila Brasilândia': 693, 'Sítio Morro Grande': 694, 'Jardim Moreira': 695, 'Jardim Andarai': 696, 'Parque São Lucas': 697, 'Jardim Mirante': 698, 'Jardim São Saverio ': 699, 'Casa Branca': 700, 'Vila São João Batista': 701, 'Jardim Santos Dumont': 702, 'Parque Panamericano': 703, 'Jardim Jussara': 704, 'Jardim São Caetano': 705, 'Jardim Monte Azul': 706, 'Jardim Caner': 707, 'Jardim Arizona': 708, 'Jardim Libano': 709, 'Vila Irmaos Arnoni': 710, 'Jardim Ester': 711, 'Vila Constanca': 712, 'Vila Laís': 713, 'Jardim Maristela': 714, 'Jardim Denise': 715, 'Lapa de Baixo': 716, 'Cidade Ademar': 717, 'Jardim Maria Duarte': 718,

'Bortolândia': 719, 'Parque Sevilha': 720, 'Vila América': 721, 'Vila Clementino ': 722, 'Santa Terezinha': 723, 'Jardim Nova Germania': 724, 'Vila Zelina': 725, 'Vila Nova das Belezas': 726, 'Jardim Cocaia': 727, 'Jardim Santa Rosa': 728, 'Vila Virginia': 729, 'Jardim Santa Cruz': 730, 'Vila Alice': 731, 'Jardim da Mamãe': 732, 'Vila Carmem': 733, 'Jardim Colombo': 734, 'Recanto Paraíso': 735, 'Vila Prel': 736, 'Vila Anadir': 737, 'Bosque da Saúde.': 738, 'Parque Santa Madalena': 739, 'Jardim Monjolo': 740, 'Chácara Cruzeiro do Sul': 741, 'Jardim Tabatinga': 742, 'Vila Santos': 743, 'Jardim Silvia': 744, 'Parque Renato Maia': 745, 'Parque Colonial': 746, 'Jardim Morumbi': 747, 'São João Clímaco': 748, 'Vila Zat': 749, 'Parque Residencial Julia': 750, 'Parque Nações Unidas': 751, 'Jardim Consórcio': 752, 'Jardim Olinda': 753, 'Jardim Jaçanã': 754, 'Vila Mafra': 755, 'Vila Portuguesa': 756, 'Jardim Kuabara': 757, 'Parque Ramos Freitas': 758, 'Vila Gopouva': 759, 'Jardim Tango': 760, 'Jardim America': 761, 'Pirituba': 762, 'Morro dos Ingleses': 763, 'Jardim Guarulhos': 764, 'Bela Aliança': 765, 'Vila Rosalia': 766, 'Chácara Tatuapé': 767, 'Jardim Vivan': 768, 'Cocaia': 769, 'Jardim Jaú (zona Leste)': 770, 'Vila America': 771, 'Jardim Cidalia': 772, 'Campininha': 773, 'Chácara Nossa Senhora do Bom Conselho': 774, 'Vila Barros': 775, 'Jardim City': 776, 'Jardim São Luís': 777, 'Cidade dos Bandeirantes': 778, 'Sítio Pinheirinho': 779, 'Jardim Sul São Paulo': 780, 'Jardim Promissão': 781, 'Vila Dionisia': 782, 'Parque Itaberaba': 783, 'Vila Pedro Moreira': 784, 'Vila Gea': 785, 'Jd. Rio Pequeno': 786, 'Vila Nova Caledonia': 787, 'Vila Prado': 788, 'Vila Harmonia': 789, 'Vila Fernandes': 790, 'Água Funda': 791, 'Vila São Francisco': 792, 'Vila Nova Esperança': 793, 'Cidade Maia': 794, 'Siciliano': 795, 'Jardim Silvestre': 796, 'Vila Londrina': 797, 'Vila Sirene': 798, 'Jardim Mimar': 799, 'Vila Dora': 800, 'Parque Santo Antônio': 801, 'Jardim da Gloria': 802, 'Vila Ernesto': 803, 'Vila Liviero': 804, 'Jardim Vazani': 805, 'Jardim das Bandeiras': 806, 'Jardim Dinorah': 807, 'Jardim Alfredo': 808, 'Jardim Haia do Carrao': 809, 'Jardim Petropolis': 810, 'Jardim Luanda': 811, 'Parque Regina': 812, 'Jardim Ana Rosa': 813, 'Vila Zamataro': 814, 'Aricanduva': 815, 'Boaçava': 816, 'Parque Continental I': 817, 'Jardim Imperador (zona Leste)': 818, 'Jardim Terezópolis': 819, 'Vila São Geraldo': 820, 'Conjunto City Jaragua': 821, 'Jardim Teresa': 822, 'Jardim Zaira': 823, 'Vila Olinda': 824, 'Parque dos Bancários': 825, 'Jardim Sônia Maria': 826, 'Jardim Carlu': 827, 'Vila Fidelis Ribeiro': 828, 'Parque Residencial da Lapa': 829, 'Vila Capitao Rabelo': 830, 'Vila Pita': 831, 'Vila Centenário': 832, 'Parque Oratório': 833, 'Vila Guarani (Zona Sul)': 834, 'Vila Gabriel': 835, 'Vila Maricy': 836, 'Jardim Santa Efigenia': 837, 'Santa Inês ': 838, 'Vila Santa Terezinha (zona Norte)': 839, 'Jardim Maria Luiza': 840, 'Jardim Itapeva': 841, 'Vila Santista': 842, 'Jardim do Carmo': 843, 'Vila Lucia': 844, 'Jardim Inga': 845, 'Santa Rita': 846, 'Vila Sao Domingos': 847, 'Jardim Santa Monica': 848, 'Parque Santos Dumont': 849, 'Horto Florestal': 850, 'Vila das Merces ': 851, 'Jardim Valeria': 852, 'Vila Progresso (zona Norte)': 853, 'Parque São Domingos': 854, 'Jardim Pirituba': 855, 'Parque Sao Domingos': 856, 'Vila Catupia': 857, 'Vila Bruna': 858, 'Vila Lúcia':

859, 'Vila Costa Melo': 860, 'Jardim Santa Bárbara': 861, 'Vila Franci': 862, 'Jardim Maria Aparecida': 863, 'Vila Fátima': 864, 'Vila Fiuza': 865, 'Jardim Bandeirantes (zona Norte)': 866, 'Jardim Santa Cecília': 867, 'Higienópolis ': 868, 'Jardim Brasil': 869, 'Vila Lucinda': 870, 'Parque Independencia': 871, 'Vila Independência': 872, 'Vila Esplanada': 873, 'Vila Hamburguesa': 874, 'Jardim Filhos da Terra': 875, 'Jardim Patente': 876, 'Cidade Brasil': 877, 'Vila Hulda': 878, 'Suísso': 879, 'Vila Santo Antônio': 880, 'Vila Bela Vista (zona Norte)': 881, 'Jardim São Judas Tadeu': 882, 'Vila Arapua': 883, 'Jardim Santo Antônio': 884, 'Interlagos': 885, 'Vila Olímpia ': 886, 'Jardim Glória': 887, 'Vila das Palmeiras': 888, 'Vila Mercedes': 889, 'Vila Ester': 890, 'Suiço': 891, 'Jardim Anny': 892, 'Vila Sao Nicolau': 893, 'Vila Antônio dos Santos': 894, 'Vila Brasilandia': 895, 'Vila Bonilha': 896, 'Jardim das Flores': 897, 'Jardim Primavera (zona Norte)': 898, 'Nova Gerty': 899, 'Parque do Estado': 900, 'Vila Granada ': 901, 'Vila Romano': 902, 'Jardim Santo Antoninho': 903, 'Vila São João': 904, 'Recanto Morumbi': 905, 'Jardim Santa Edwirges': 906, 'Vila Maria Zélia': 907, 'Vila Matiilde': 908, 'Jardim America da Penha': 909, 'Agua Branca': 910, 'Vila Cavaton': 911, 'Jardim Dona Sinha': 912, 'Parque Penha': 913, 'Jardim Sao Jose': 914, 'Itaquaciara': 915, 'Jardim Odete': 916, 'Vila São Judas Tadeu': 917, 'Jardim Monte Libano': 918, 'Jordanópolis': 919, 'Vila Tijuco': 920, 'Conjunto Residencial Butantã': 921, 'Jardim Franca': 922, 'Chácara Santo Antônio': 923, 'Chácara Itaim': 924, 'Jardim Panorama (zona Leste)': 925, 'Jardim Maia': 926, 'Jardim Andaraí': 927, 'Cupecê': 928, 'Vila Amalia (zona Leste)': 929, 'Jardim Santa Inês': 930, 'Vila Campo Grande': 931, 'Vila Nelson': 932, 'Vila Irmãos Arnoni': 933, 'Belém': 934, 'Jardim das Nações': 935, 'Jardim Maringa': 936, 'Vila Metalúrgica': 937, 'Jardim Vergueiro': 938, 'Vila Praia': 939, 'Vila do Bosque': 940, 'Jardim Nossa Senhora Aparecida': 941, 'Barro Branco (zona Norte)': 942, 'Vila Cristalia': 943, 'Varzea do Glicerio': 944, 'Centro Capital': 945, 'Bras': 946, 'Vila Nogueira': 947, 'Vila Lúcia Elvira': 948, 'Vila Lucia Elvira': 949, 'Jardim Analia Franco': 950, 'Vila Diva': 951, 'Lar São Paulo': 952, 'Jardim Lar Sao Paulo': 953, 'Jardim Taboao': 954, 'Jardim Morro Verde': 955, 'Vila Albano': 956, 'Jardim Frei Galvão': 957, 'Jardim Frei Galvao': 958, 'Parque Bairro Morumbi': 959, 'Jardim Dracena': 960, 'Jardim das Esmeraldas': 961, 'Jardim Nadir': 962, 'Jardim Guarau': 963, 'Fazenda Morumbi ': 964, 'Jardim Luísa': 965, 'Jardim Monte Alegre': 966, 'Jardim Maria Rosa': 967, 'Morumbi ': 968, 'Chácara Agrindus': 969, 'Raposo Tavares': 970, 'Jardim Novo Taboao': 971, 'Jardim Rosa Maria': 972, 'Jardim Arpoador': 973, 'Vila Tiradentes': 974, 'Vila Polopoli': 975, 'Jardim Ester Yolanda': 976, 'Parque Monte Alegre': 977, 'Jardim América': 978, 'Jardim Adhemar de Barros': 979, 'Parque Esmeralda': 980, 'Vila Antonio': 981, 'Jardim Lucia': 982, 'Nossa Senhora do O': 983, 'Vila São Vicente': 984, 'Freguesia do Ó ': 985, 'Vila União': 986, 'Vila Arcádia': 987, 'Jardim das Graças': 988, 'Vila Carolina': 989, 'Bela Aliança ': 990, 'Vila Picinin': 991, 'Parque Mandi': 992, 'Jardim São Silvestre': 993, 'Jardim Cachoeira': 994, 'Vila Hermínia': 995, 'Vila Barreto': 996,

'Vila Fiat Lux': 997, 'Jardim Felicidade (zona Oeste)': 998, 'Paraíso.': 999, 'Jardim dos Estados': 1000, 'Jardim Cordeiro': 1001, 'Jardim Los Angeles': 1002, 'Jardins': 1003, 'Vila Campesina': 1004, 'São Pedro': 1005, 'Jaguaribe': 1006, 'Jardim João Xxiii': 1007, 'Bussocaba': 1008, 'Veloso': 1009, 'Jardim Roberto': 1010, 'Km 18': 1011, 'Cipava': 1012, 'Conceição': 1013, 'Cidade Intercap': 1014, 'Piratininga': 1015, 'Umuarama': 1016, 'Presidente Altino': 1017, 'Quitaúna': 1018, 'Jardim Cirino': 1019, 'Vila Osasco': 1020, 'Padroeira': 1021, 'Novo Osasco': 1022, 'Vila Yolanda': 1023, 'Vila Yara': 1024, 'Cidade das Flores': 1025, 'City Bussocaba': 1026, 'Pestana': 1027, 'Parque dos Principes': 1028, 'Jardim Amaralina': 1029, 'Vila Antônio': 1030, 'Bandeiras': 1031, 'Jardim D'abril': 1032, 'Ayrosa': 1033, 'Vila Adalgisa': 1034, 'Bonfim': 1035, 'Adalgisa': 1036, 'Cidade Jardim': 1037, 'Helena Maria': 1038, 'Jardim Gilda Maria': 1039, 'Conceicao': 1040, 'Jardim Elvira': 1041, 'Industrial Autonomistas': 1042, 'Jardim Munhoz Junior': 1043, 'I.a.p.i.': 1044, 'Mutinga': 1045, 'Recanto das Rosas': 1046, 'Jardim Sarah': 1047, 'Parque Ipe': 1048, 'Vila Comercial': 1049, 'Parque Ipê': 1050, 'Jardim Agu': 1051, 'Jardim Primeiro de Maio (chacara Fazendinha)': 1052, 'Vila Dalva': 1053, 'Jardim Vista Alegre': 1054, 'Cidade Líder': 1055, 'Colônia (zona Leste)': 1056, 'Vila Campanela': 1057, 'Jardim Santa Terezinha (zona Leste)': 1058, 'Jardim Penha': 1059, 'Parque das Paineiras': 1060, 'Jardim Nova Cidade': 1061, 'Vila Paranagua': 1062, 'Itaquera': 1063, 'Jardim Pedro José Nunes': 1064, 'Vila Nova Curuçá': 1065, 'Fazenda Aricanduva': 1066, 'Vila Carmosina': 1067, 'Jardim Helian': 1068, 'Parque Cecap': 1069, 'Conjunto Residencial José Bonifácio': 1070, 'Jardim Belém': 1071, 'Parque Cisper': 1072, 'Jardim Nordeste': 1073, 'Parque Sao Francisco': 1074, 'Conjunto Habitacional Padre Manoel da Nóbrega': 1075, 'Cidade Antônio Estêvão de Carvalho': 1076, 'Jardim Norma': 1077, 'Parque Guarani': 1078, 'Cidade Satelite Santa Barbara': 1079, 'Cidade Antônio Estevão de Carvalho': 1080, 'Vila Princesa Isabel': 1081, 'Vila Cosmopolita': 1082, 'Jardim Brasília (zona Leste)': 1083, 'Vila Jacuí': 1084, 'Vila Nhocune': 1085, 'Artur Alvim': 1086, 'Vila Santa Teresa (zona Leste)': 1087, 'Parque Cruzeiro do Sul': 1088, 'Ermelino Matarazzo': 1089, 'Parque Artur Alvim': 1090, 'São Miguel Paulista': 1091, 'Conjunto Promorar Sapopemba': 1092, 'Cidade Nova São Miguel': 1093, 'Vila Rio Branco': 1094, 'Parque Boturussu': 1095, 'Guaianazes': 1096, 'Jardim Santo Antonio': 1097, 'Jardim Helena': 1098, 'Parque Paineiras': 1099, 'Jardim Marília': 1100, 'Vila Paranaguá': 1101, 'Jardim Vila Carrao': 1102, 'Jardim Sao Jose (sao Mateus)': 1103, 'Burgo Paulista': 1104, 'Vila São Francisco (zona Leste)': 1105, 'Jardim Sao Gabriel': 1106, 'Conjunto Habitacional Padre Manoel da Nobrega': 1107, 'Jardim Tietê': 1108, 'Jardim Lideranca': 1109, 'Cidade São Mateus': 1110, 'Parada Xv de Novembro': 1111, 'Jardim do Colégio (zona Norte)': 1112, 'Jardim Sao Bento': 1113, 'Vila Aurora (Zona Norte)': 1114, 'Vila Pauliceia': 1115, 'Usina Piratininga': 1116, 'Jardim Primavera (zona Sul)': 1117, 'Jardim Satelite': 1118, 'Jardim Palmares (zona Sul)': 1119, 'Jardim Sabara': 1120, 'Vila Castelo': 1121, 'Pedreira': 1122, 'Vila da Paz': 1123, 'Jardim Campo Grande': 1124, 'Capela do Socorro': 1125, 'Jardim

```
Ernestina': 1126, 'Terceira Divisão de Interlagos': 1127, 'Jardim
Ipanema (zona Sul)': 1128, 'Cidade Dutra': 1129, 'Jardim Santa Cruz
(campo Grande)': 1130, 'Jardim Lallo': 1131, 'Vila Anhanguera': 1132,
'Vila Arriete': 1133, 'Jardim Regis': 1134, 'Jardim Sabará': 1135,
'Veleiros': 1136, 'Jardim Jua': 1137, 'Vila Emir': 1138, 'Vila
Anhangüera': 1139, 'Jardim Maraba': 1140, 'Jardim Três Marias': 1141,
'Jardim dos Lagos': 1142, 'Vila Friburgo': 1143, 'Jardim Ana Lúcia':
1144, 'Chácara Meyer': 1145, 'Vila California(zona Sul)': 1146,
'Jardim Bélgica': 1147, 'Jardim Ubirajara': 1148, 'Jardim Santa
Helena': 1149, 'Jardim Marajoara ': 1150, 'Jardim Guarapiranga': 1151,
'Vila Califórnia(zona Sul)': 1152, 'Jardim Cristal': 1153, 'Jardim da
Campina': 1154, 'Jardim Maria Rita': 1155, 'Bolsão do Interlagos':
1156, 'Jardim dos Prados': 1157, 'Parque Munhoz': 1158, 'Pacaembu.':
1159, 'Vila Boa Vista': 1160, 'Vila Engenho Novo': 1161, 'Jardim
Tupanci': 1162, 'Alphaville Empresarial': 1163, 'Alphaville': 1164,
'Nova Aldeinha': 1165, 'Tamboré': 1166, 'Alphaville Residencial Um':
1167, 'Parque Santa Luzia': 1168, 'Empresarial 18 do Forte': 1169,
'Alphaville Industrial': 1170, 'Melville Empresarial I E Ii': 1171,
'Residencial Tres (tambore)': 1172, 'Melville Empresarial Ii': 1173,
'Centro Empresarial Tamboré': 1174, 'Vila Morellato': 1175, 'Vila Sao
Luiz (valparaizo)': 1176, 'Alphaville Conde Ii': 1177, 'Alphaville
Centro Industrial E Empresarial/alphaville.': 1178, 'Bethaville I':
1179, 'Jardim dos Camargos': 1180, 'Jardim Barueri': 1181, 'Jardim
Paraiso': 1182, 'Centro Comercial Jubran': 1183, 'Residencial Cinco
(alphaville)': 1184, 'Jardim Graziela': 1185, 'Centro de Apoio I
(alphaville)': 1186, 'Jardim Regina Alice': 1187, 'Alphaville
Residencial Plus': 1188, 'Vila Pouso Alegre': 1189, 'Alphaville
Residencial Dois': 1190, 'Vila Conceição': 1191, 'Residencial Seis
(alphaville)': 1192, 'Vila Sargento José de Paula': 1193, 'Parque do
Morumbi': 1194, 'Jardim Vitoria Regia (zona Oeste)': 1195, 'Vila São
Francisco (zona Sul)': 1196, 'Vila Elvira': 1197, 'Jardim Vitoria
Regia': 1198, 'Retiro Morumbi': 1199}
```

```
df_prep1['Distrito_ID'] = df_prep1['Distrito'].map(distrito_id)
```

```
df_prep1.head()
```

```
       Distrito  Area  Quartos  Garagem                Tipo  Aluguel
IPTU  \
0    Belenzinho    21        1        0     Studio e kitnet     2400
539
1  Vila Marieta    15        1        1     Studio e kitnet     1030
315
2     Pinheiros    18        1        0         Apartamento     4000
661
3       Vila Ré    56        2        2  Casa em condomínio     1750
204
4    Bela Vista    19        1        0     Studio e kitnet     4000
654
```

```
   Tipo_ID  Distrito_ID
0        1            1
1        1            2
2        2            3
3        3            4
4        1            5
```

```
df_prep1.shape
```

```
(11618, 9)
```

```
colunas_float = ['Area', 'Aluguel', 'IPTU']
colunas_int = ['Quartos', 'Garagem', 'Distrito_ID', 'Tipo_ID']

df_prep1[colunas_float] = df_prep1[colunas_float].astype(float)
df_prep1[colunas_int] = df_prep1[colunas_int].astype(int)
df_prep1.dtypes
```

```
Distrito        object
Area           float64
Quartos          int32
Garagem          int32
Tipo            object
Aluguel        float64
IPTU           float64
Tipo_ID          int32
Distrito_ID      int32
dtype: object
```

```
df_prep2 = df_prep1.copy()
df_prep2.head()
```

```
       Distrito  Area  Quartos  Garagem                Tipo  Aluguel
IPTU  \
0    Belenzinho  21.0        1        0     Studio e kitnet   2400.0
539.0
1  Vila Marieta  15.0        1        1     Studio e kitnet   1030.0
315.0
2     Pinheiros  18.0        1        0         Apartamento   4000.0
661.0
3       Vila Ré  56.0        2        2  Casa em condomínio   1750.0
204.0
4    Bela Vista  19.0        1        0     Studio e kitnet   4000.0
654.0
```

```
   Tipo_ID  Distrito_ID
0        1            1
1        1            2
2        2            3
3        3            4
4        1            5
```

```
zn_centro = ['Bela Vista', 'Bom Retiro', 'Brás', 'Cambuci',
'Consolação', 'Liberdade', 'Pari', 'República', 'Santa Cecília', 'Sé']

zn_norte = ['Anhanguera', 'Brasilândia', 'Cachoeirinha', 'Casa Verde',
'Freguesia do Ó', 'Jacana', 'Jaçanã', 'Jaraguá', 'Limão', 'Perus',
'Santana', 'Tremembé', 'Tucuruvi', 'Vila Guilherme', 'Vila Maria',
'Vila Medeiros']

zn_sul = ['Campo Belo', 'Campo Grande', 'Campo Limpo', 'Capão
Redondo', 'Cidade Ademar', 'Cidade Dutra', 'Cidade Líder', 'Cidade
Tiradentes', 'Grajaú', 'Ipiranga', 'Jabaquara', 'Jardim Ângela',
'Jardim São Luís', 'Jardim Paulista', 'Jardim Helena', 'Marsilac',
'Moema', 'Mooca', 'Morumbi', 'Parelheiros', 'Pedreira', 'Sacomã',
'Santo Amaro', 'Socorro', 'São Lucas', 'São Mateus', 'São Rafael',
'Sapopemba', 'Saúde', 'Vila Andrade', 'Vila Mariana', 'Vila Mascote',
'Vila Olímpia', 'Vila Sonia']

zn_leste = ['Água Rasa', 'Aricanduva', 'Artur Alvim', 'Belém',
'Cangaíba', 'Carrão', 'Ermelino Matarazzo', 'Guaianases', 'Iguatemi',
'Itaim Paulista', 'Itaquera', 'Jardim Helena', 'José Bonifácio',
'Lageado', 'Penha', 'Ponte Rasa', 'São Miguel', 'São Mateus', 'São
Rafael', 'São Lucas', 'Sapopemba', 'Tatuapé', 'Vila Curuçá', 'Vila
Esperança', 'Vila Formosa']

zn_oeste = ['Alto de Pinheiros', 'Barra Funda', 'Butantã', 'Jaguara',
'Jaguaré', 'Lapa', 'Pacaembu', 'Perdizes', 'Pinheiros', 'Pirituba',
'Raposo Tavares', 'Rio Pequeno', 'Vila Leopoldina']


dist_ofc = zn_centro + zn_leste + zn_norte + zn_oeste + zn_sul

distrito_zona_map = {}

for distrito in zn_centro:
    distrito_zona_map[distrito] = 'Centro'
for distrito in zn_leste:
    distrito_zona_map[distrito] = 'Leste'
for distrito in zn_norte:
    distrito_zona_map[distrito] = 'Norte'
for distrito in zn_oeste:
    distrito_zona_map[distrito] = 'Oeste'
for distrito in zn_sul:
    distrito_zona_map[distrito] = 'Sul'

df_prep2['Zonas'] = df_prep2['Distrito'].map(distrito_zona_map)

zonas_unicas = df_prep2['Zonas'].unique()
zonas_id = {zonas: idx for idx, zonas in enumerate(zonas_unicas,
start=1)}
print(zonas_id)
```

```
{nan: 1, 'Oeste': 2, 'Centro': 3, 'Leste': 4, 'Sul': 5, 'Norte': 6}
```

```
df_prep2['Zonas_ID'] = df_prep2['Zonas'].map(zonas_id)
```

```
df_prep2['Zonas'].unique()
```

```
array([nan, 'Oeste', 'Centro', 'Leste', 'Sul', 'Norte'], dtype=object)
```

```
df_prep2.head()
```

```
       Distrito  Area  Quartos  Garagem                Tipo  Aluguel
IPTU  \
0    Belenzinho  21.0        1        0     Studio e kitnet   2400.0
539.0
1  Vila Marieta  15.0        1        1     Studio e kitnet   1030.0
315.0
2     Pinheiros  18.0        1        0         Apartamento   4000.0
661.0
3       Vila Ré  56.0        2        2  Casa em condomínio   1750.0
204.0
4    Bela Vista  19.0        1        0     Studio e kitnet   4000.0
654.0

   Tipo_ID  Distrito_ID   Zonas  Zonas_ID
0        1            1     NaN         1
1        1            2     NaN         1
2        2            3   Oeste         2
3        3            4     NaN         1
4        1            5  Centro         3
```

```
df_dist_ofc = df_prep2[df_prep2["Distrito"].isin(dist_ofc)]
print(df_dist_ofc.shape)
df_dist_ofc.head()
```

```
(3689, 11)
```

```
      Distrito  Area  Quartos  Garagem             Tipo  Aluguel
IPTU  \
2    Pinheiros  18.0        1        0      Apartamento   4000.0
661.0
4   Bela Vista  19.0        1        0  Studio e kitnet   4000.0
654.0
5         Brás  50.0        2        1      Apartamento   3800.0
787.0
10          Sé  63.0        2        0      Apartamento   1500.0
520.0
11          Sé  34.0        1        0      Apartamento   1000.0
406.0

    Tipo_ID  Distrito_ID   Zonas  Zonas_ID
2         2            3   Oeste         2
```

```
4         1            5  Centro         3
5         2            6  Centro         3
10        2           11  Centro         3
11        2           11  Centro         3
```

```
df_dist_ofc.describe()
```

```
              Area      Quartos      Garagem       Aluguel
IPTU  \
count  3689.000000  3689.000000  3689.000000   3689.000000
3689.000000
mean     77.014096     1.838439     0.894280   3500.051504
985.356194
std      67.918389     0.915300     1.015975   2667.434023
1012.698053
min       1.000000     1.000000     0.000000    504.000000
13.000000
25%      37.000000     1.000000     0.000000   1850.000000
429.000000
50%      55.000000     2.000000     1.000000   2700.000000
695.000000
75%      87.000000     2.000000     1.000000   4000.000000
1174.000000
max     580.000000     6.000000     6.000000  15000.000000
11550.000000

           Tipo_ID  Distrito_ID     Zonas_ID
count  3689.000000  3689.000000  3689.000000
mean      2.089455   102.399024     3.972079
std       0.825384   143.775817     1.288415
min       1.000000     3.000000     2.000000
25%       2.000000    16.000000     3.000000
50%       2.000000    87.000000     4.000000
75%       2.000000   142.000000     5.000000
max       4.000000  1129.000000     6.000000
```

```
df_dist_ofc['Distrito_ID'].value_counts()
```

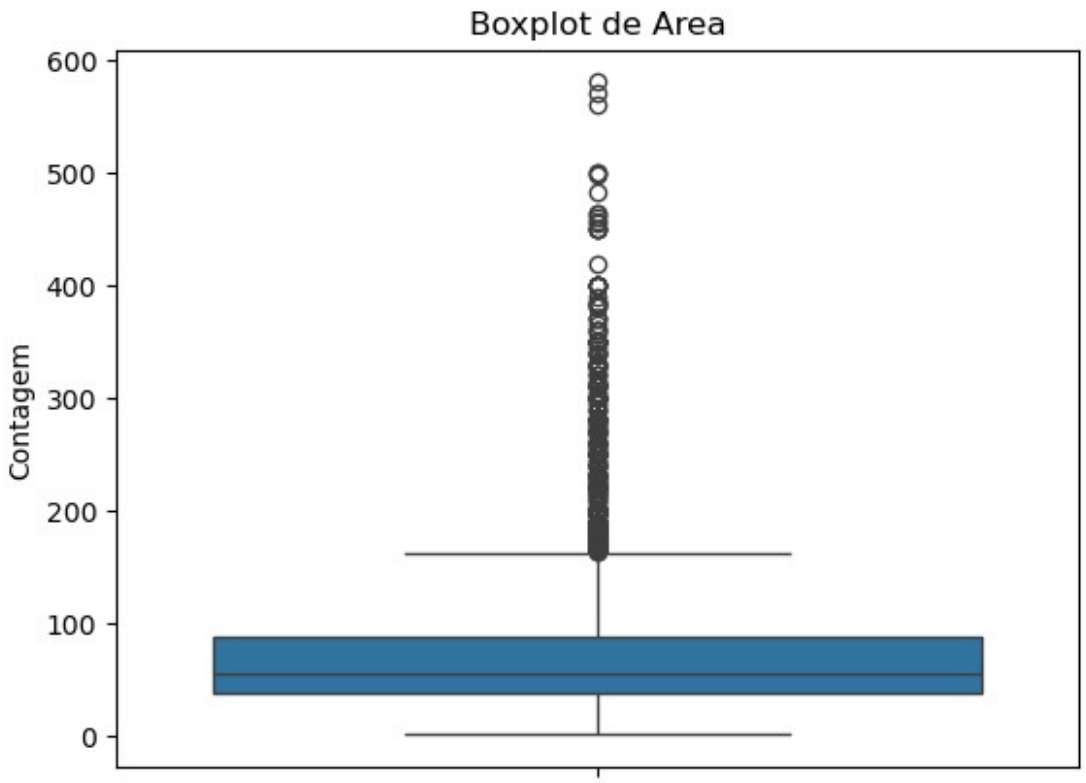
```
Distrito_ID
5       350
102     232
152     220
3       159
87      155
       ...
634       1
815       1
934       1
970       1
```

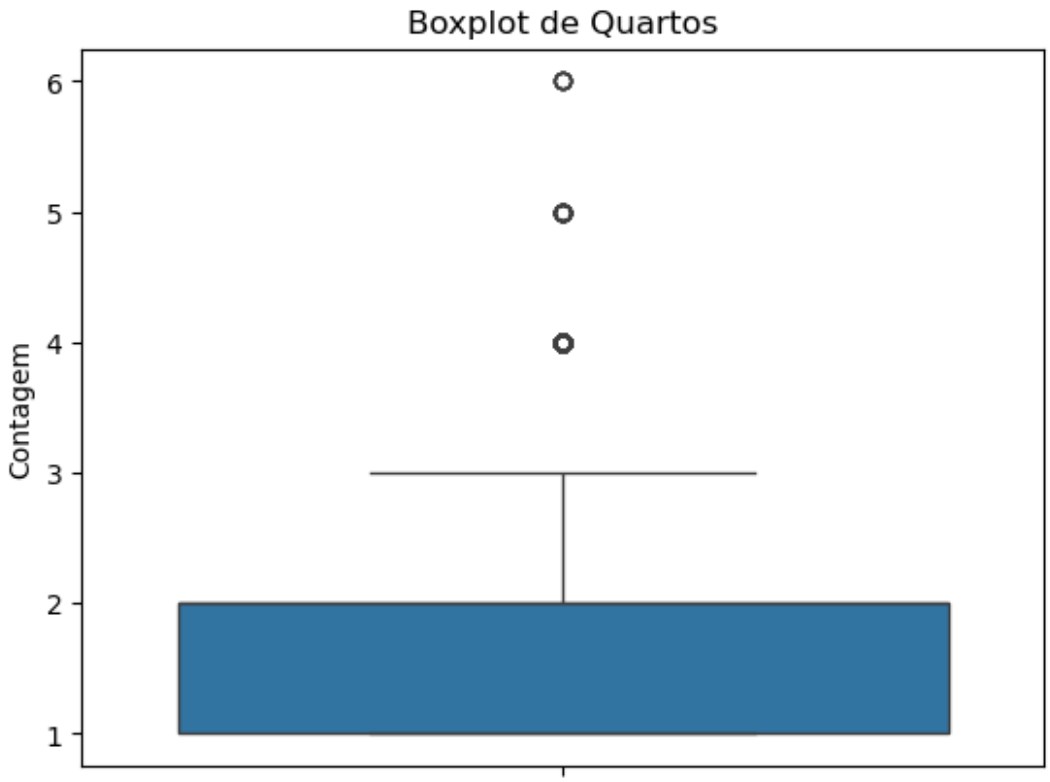
```
1098      1
Name: count, Length: 70, dtype: int64
```

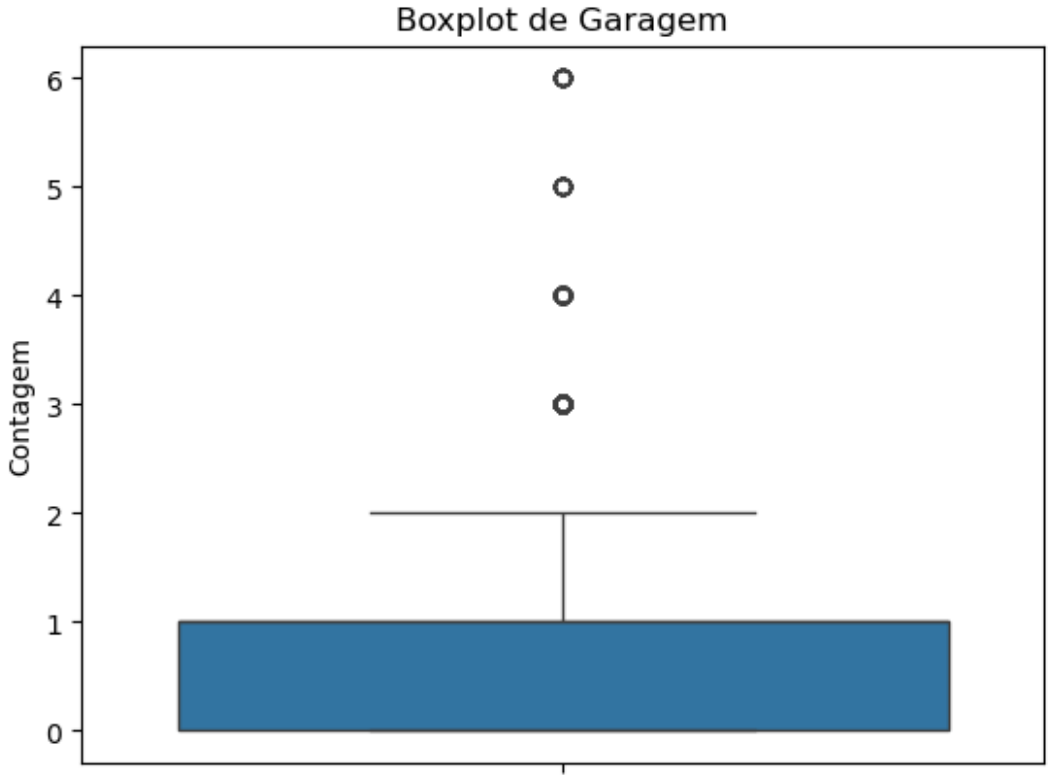
```
df_dist_ofc['Zonas_ID'].value_counts()
```

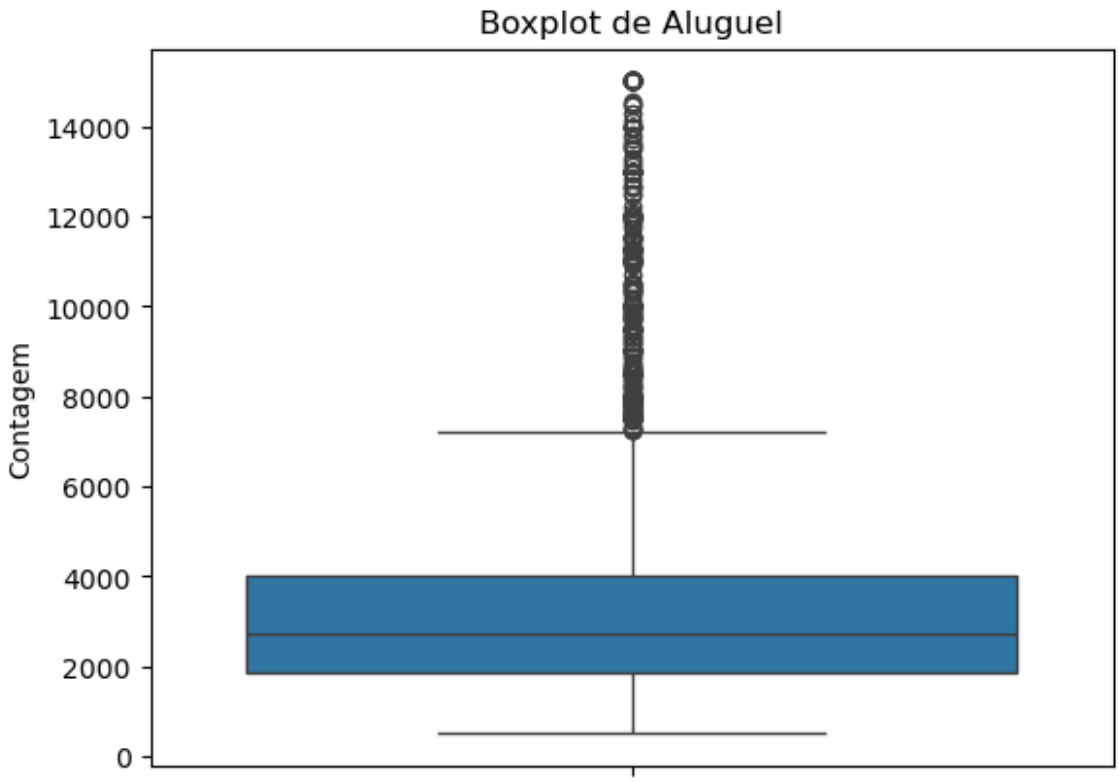
```
Zonas_ID
5    1413
3    1152
2     536
6     354
4     234
Name: count, dtype: int64
```

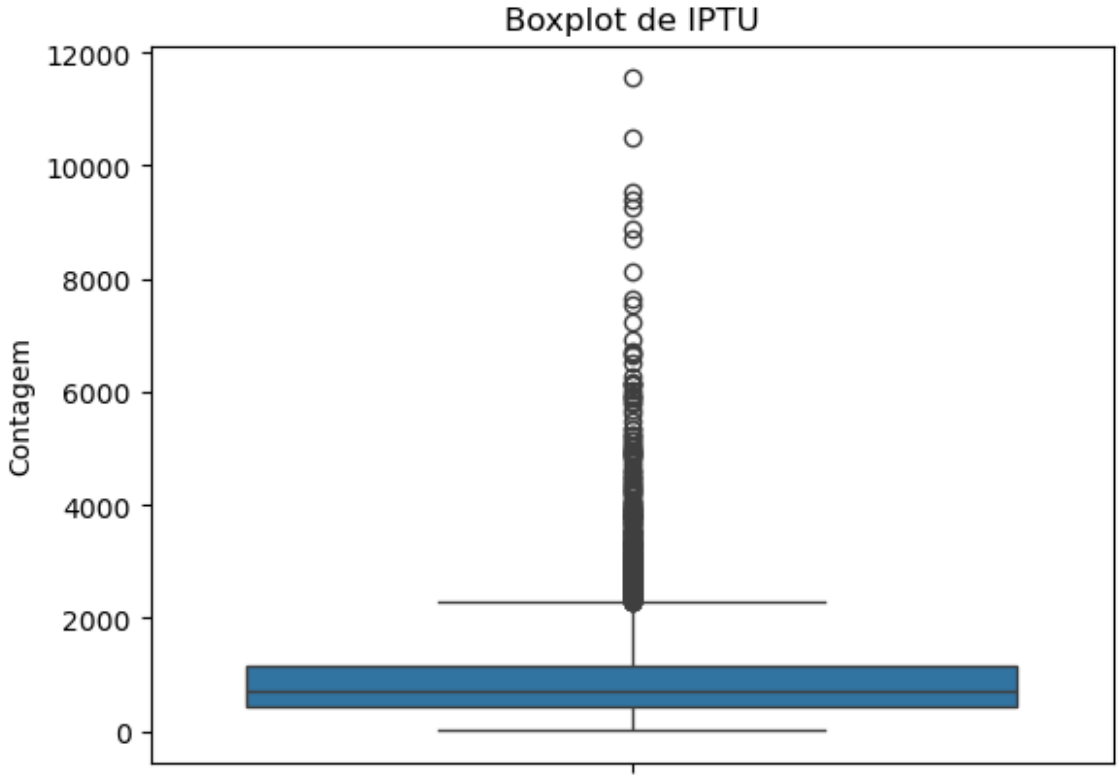
```
df_colunas = ['Area', 'Quartos', 'Garagem', 'Aluguel', 'IPTU',
'Tipo_ID',  'Distrito_ID', 'Zonas_ID']

for column in df_colunas:
    sns.boxplot(df_dist_ofc[column])
    plt.title(f'Boxplot de {column}')
    plt.ylabel('Contagem')
    plt.show()
```

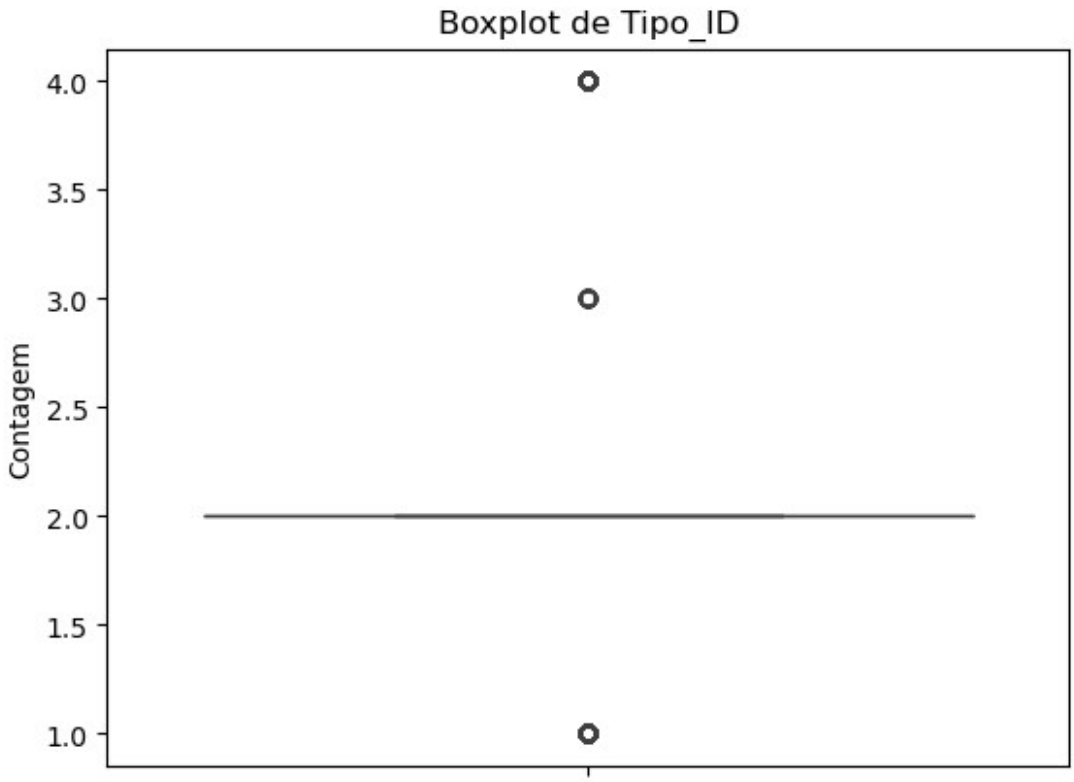


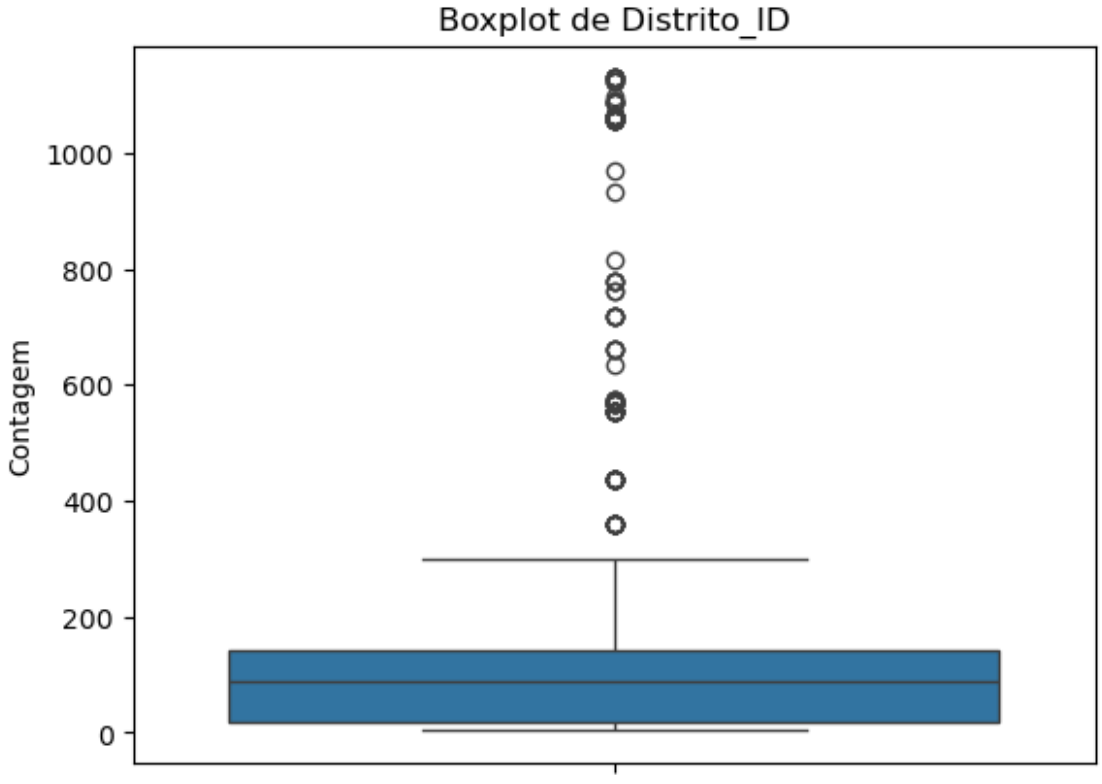

Boxplot de Quartos
Contagem
6
5
4
3
2
1

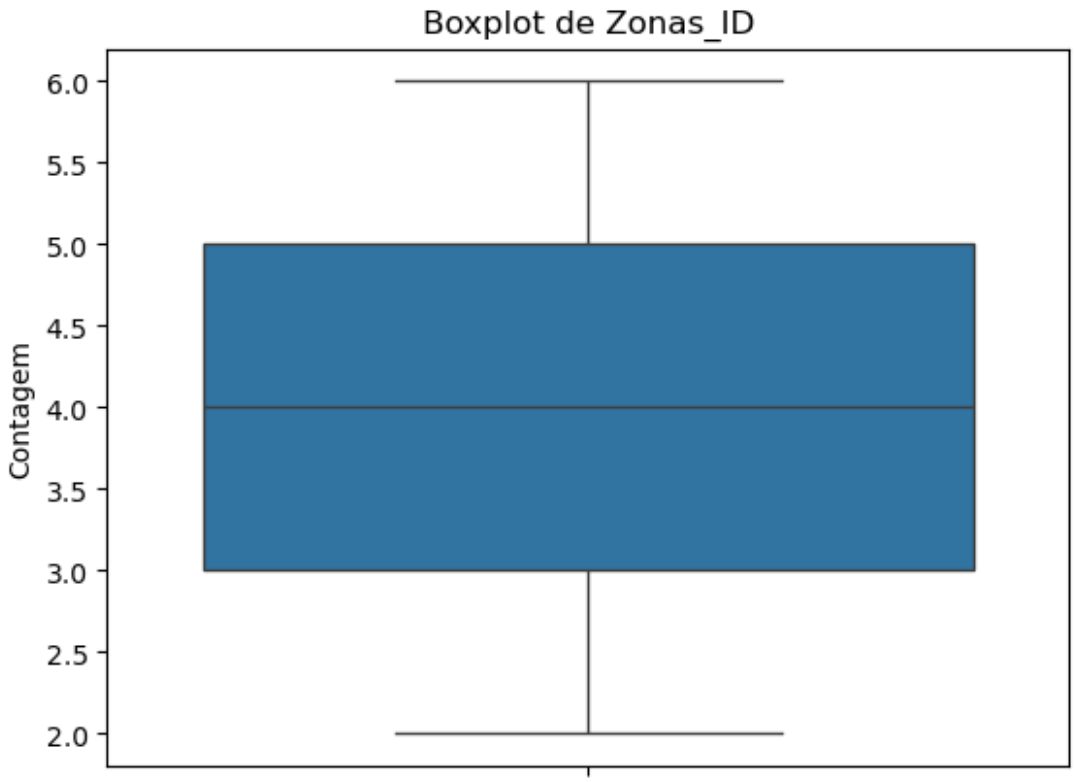

Boxplot de Garagem
Contagem
6
5
4
3
2
1
0


Boxplot de Aluguel
Contagem
14000
12000
10000
8000
6000
4000
2000
0

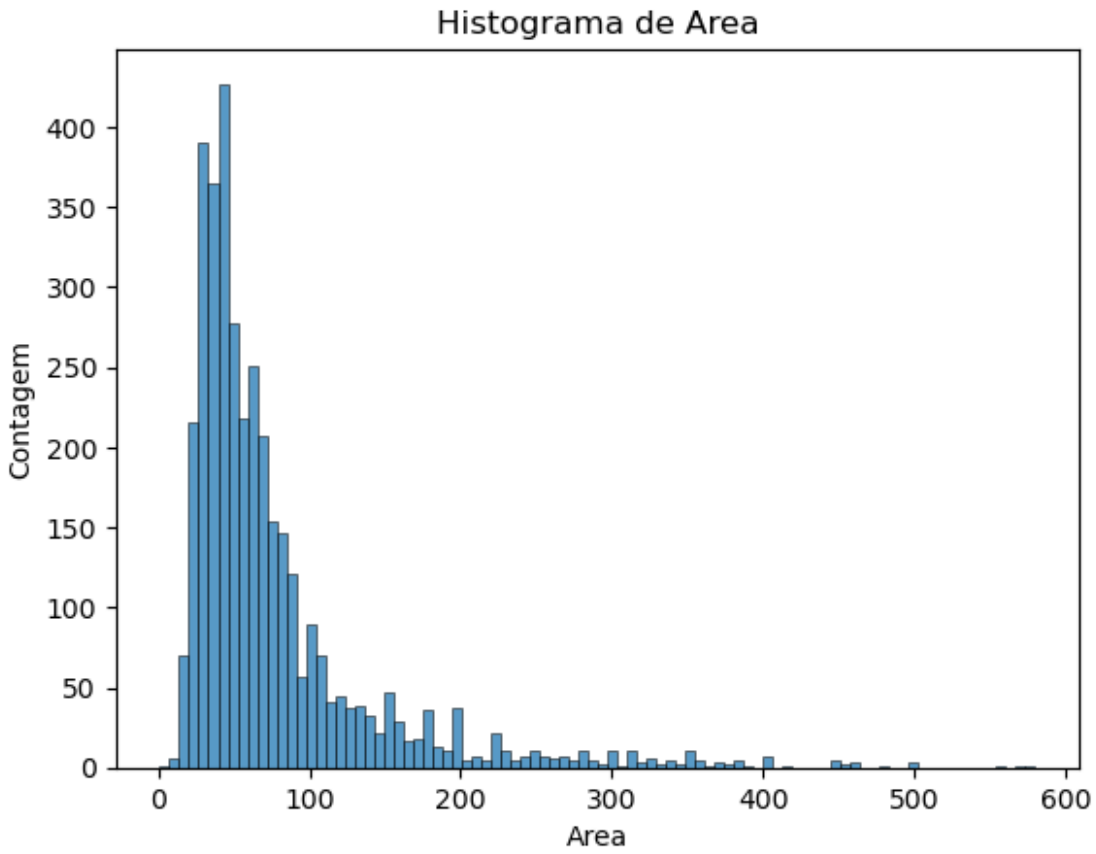

Boxplot de IPTU
Contagem
12000
10000
8000
6000
4000
2000
0

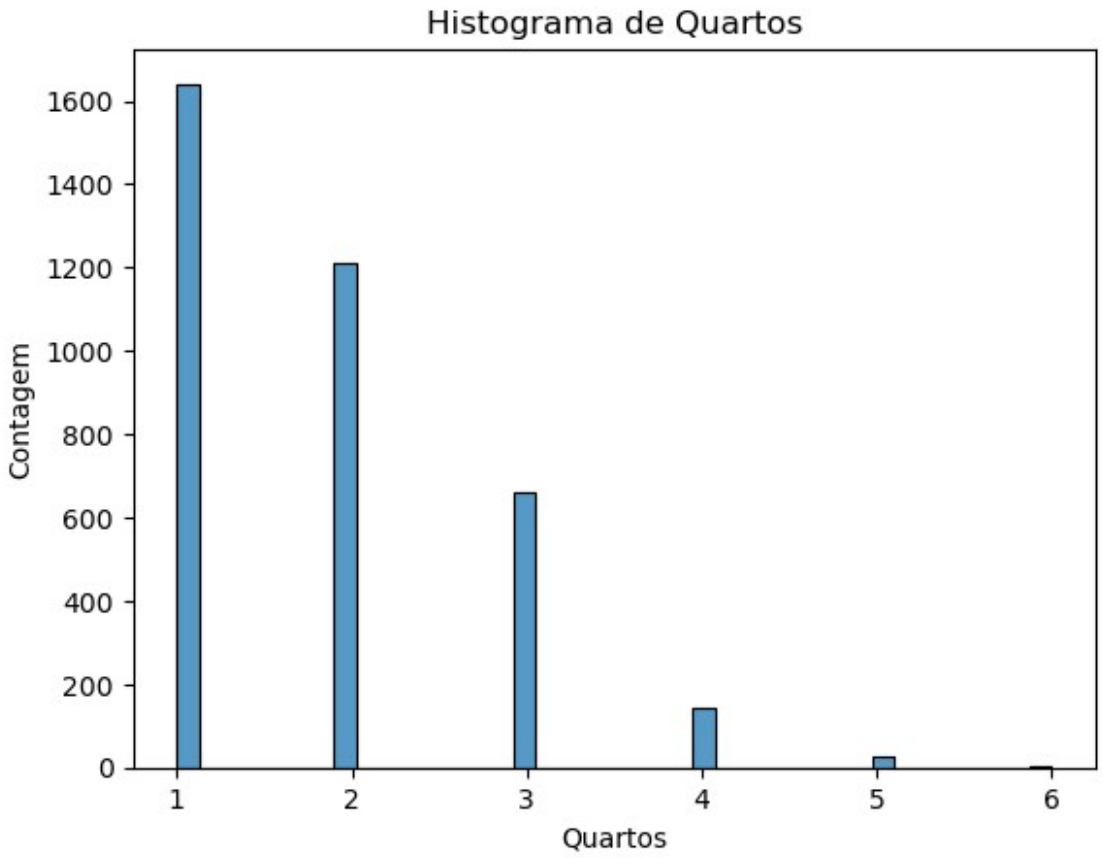

Boxplot de Tipo_ID
Contagem
4.0
3.5
3.0
2.5
2.0
1.5
1.0

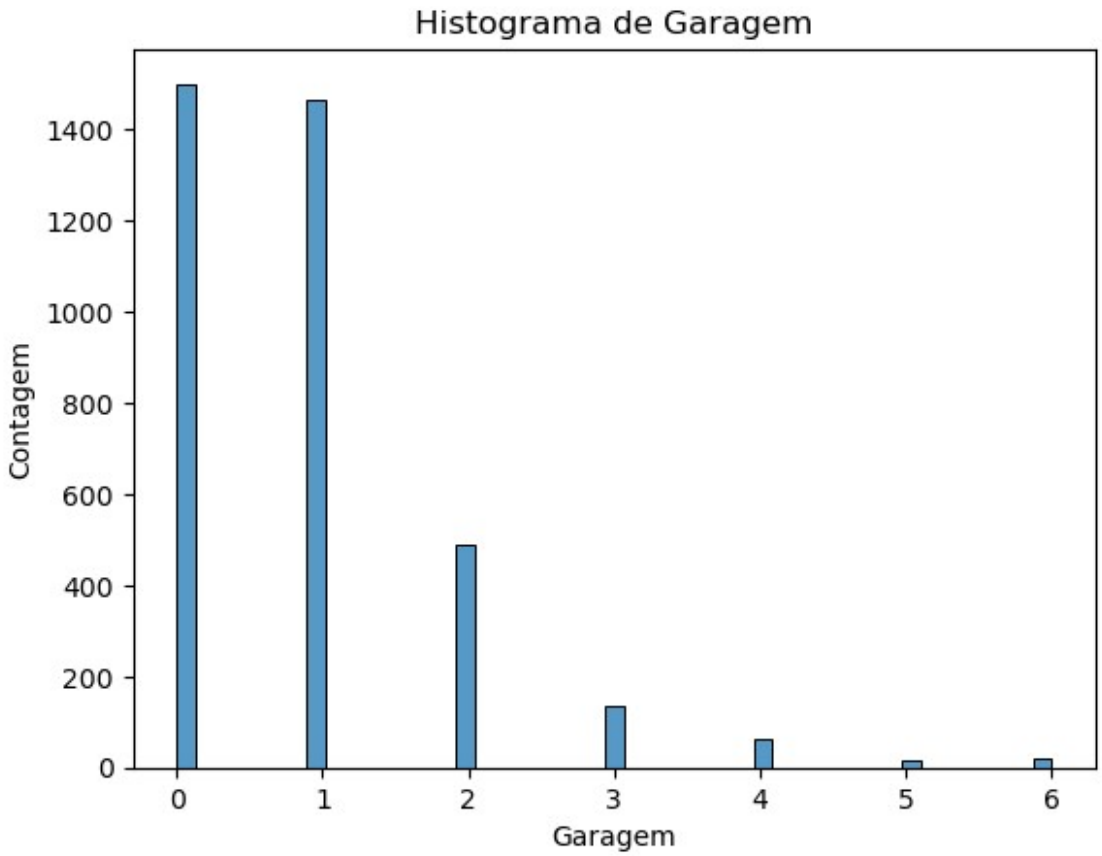

Boxplot de Distrito_ID
Contagem
1000
800
600
400
200
0

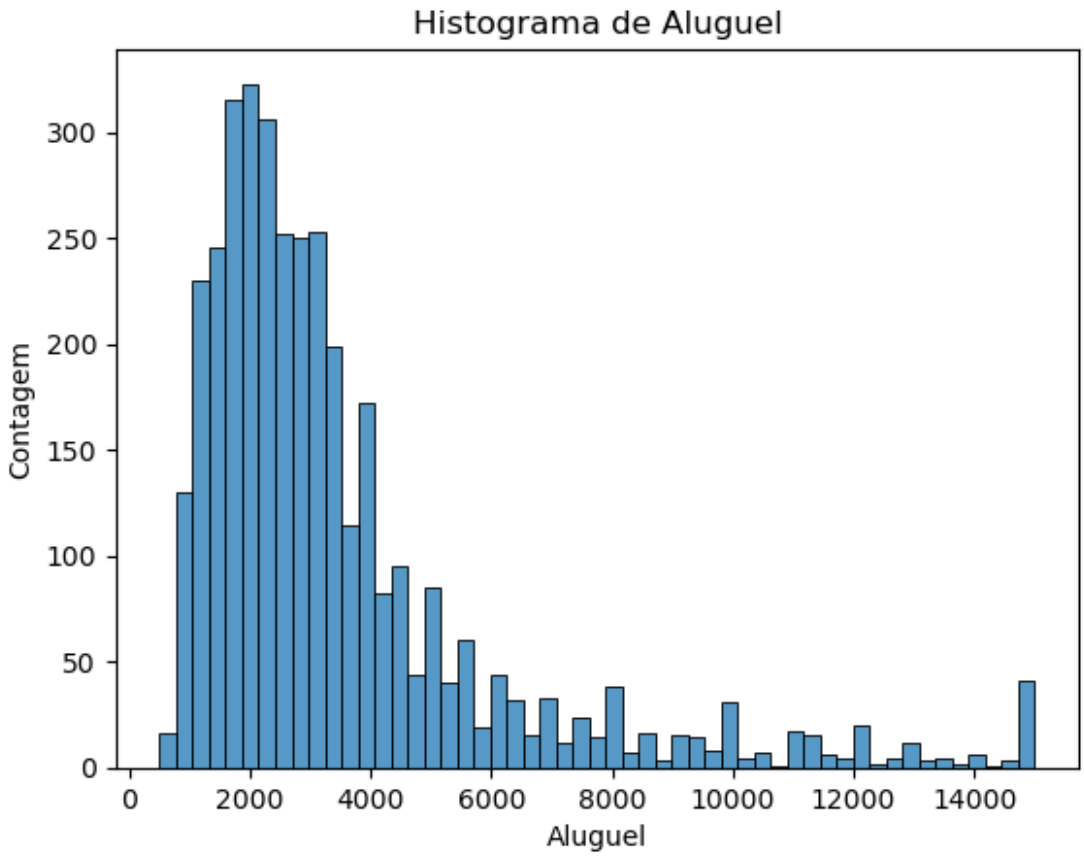


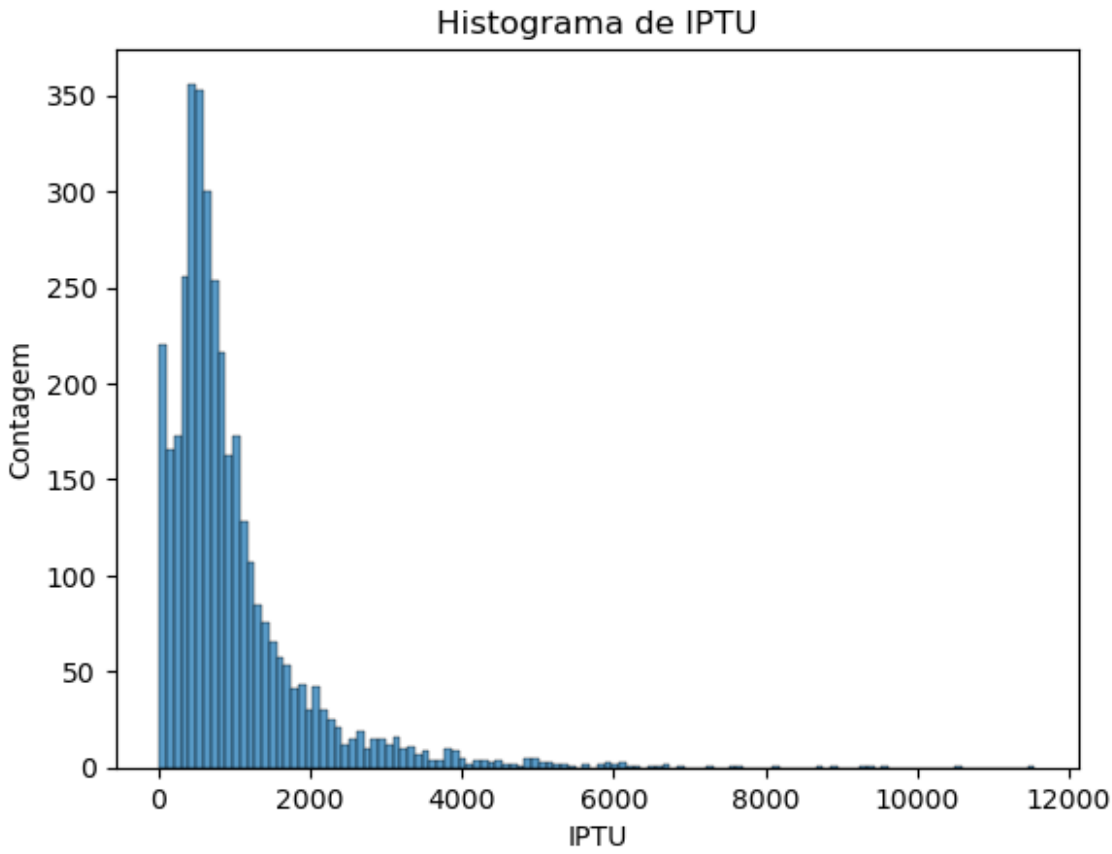

```
for column in df_colunas:
    sns.histplot(df_dist_ofc[column])
    plt.title(f'Histograma de {column}')
    plt.ylabel('Contagem')
    plt.show()
```

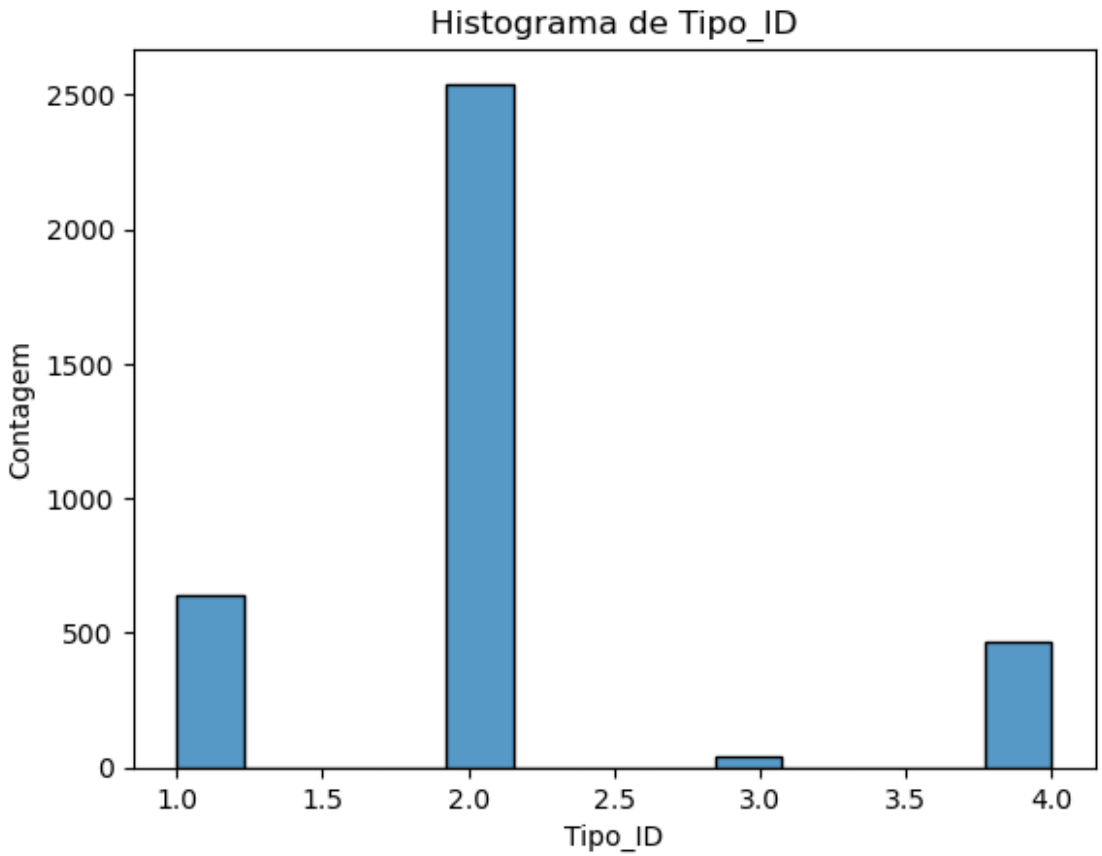

Histograma de Area
Contagem
Area
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600

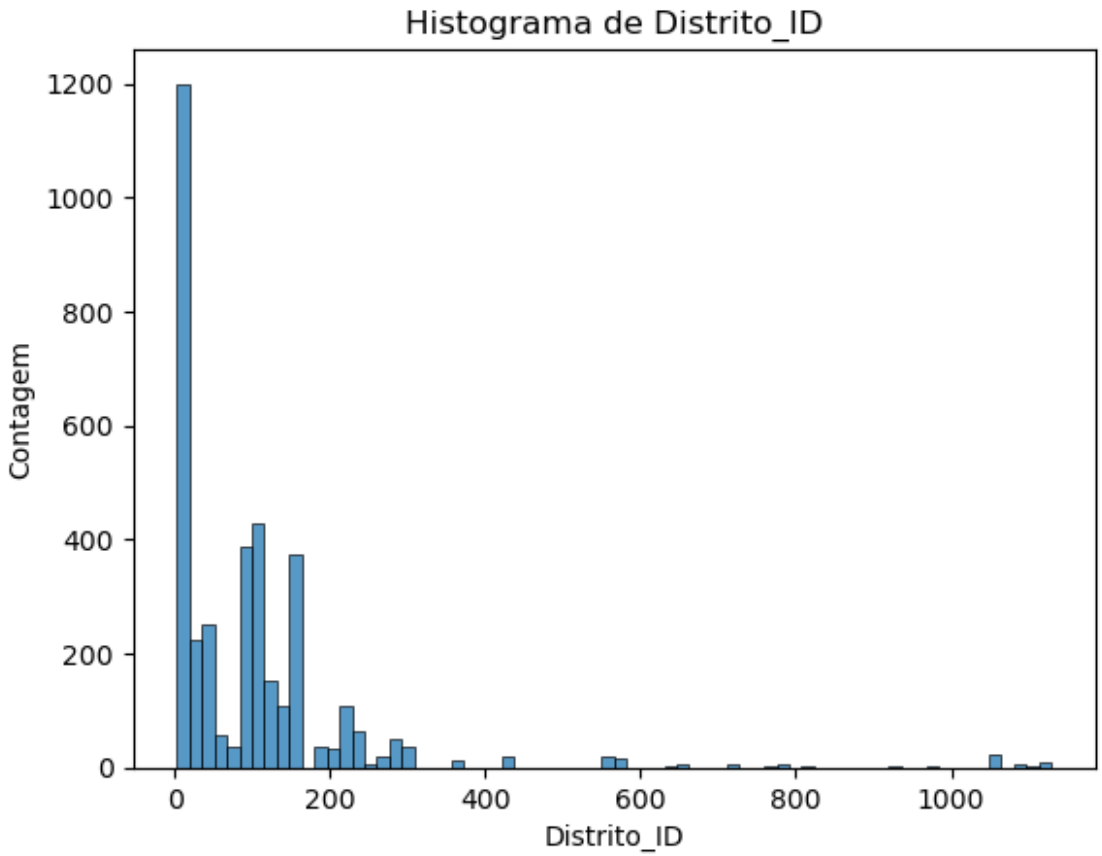

Histograma de Quartos
Contagem
Quartos
0
200
400
600
800
1000
1200
1400
1600
1
2
3
4
5
6

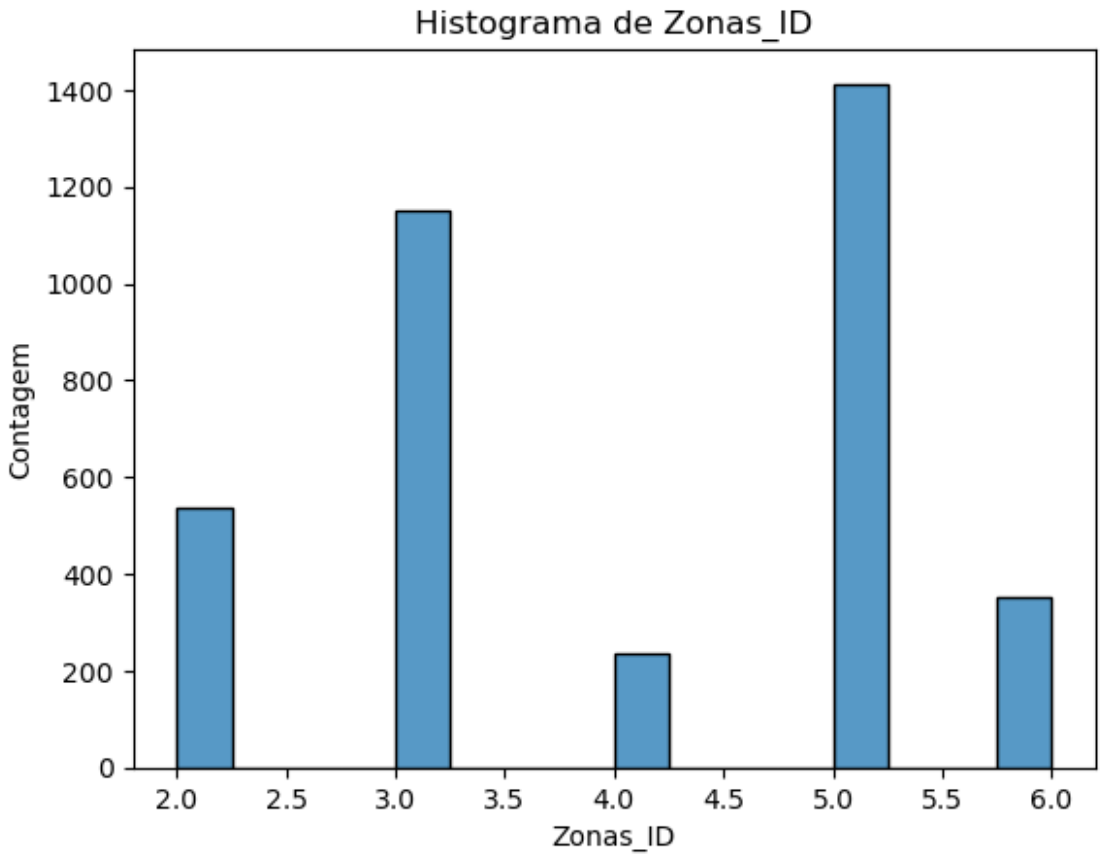

Histograma de Garagem
Contagem
Garagem
0
200
400
600
800
1000
1200
1400
0
1
2
3
4
5
6


Histograma de Aluguel
Contagem
Aluguel
0
50
100
150
200
250
300
0
2000
4000
6000
8000
10000
12000
14000


Histograma de IPTU
Contagem
IPTU
0
50
100
150
200
250
300
350
0
2000
4000
6000
8000
10000
12000


Histograma de Tipo_ID
Contagem
Tipo_ID
0
500
1000
1500
2000
2500
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Histograma de Distrito_ID
Contagem
Distrito_ID
0
200
400
600
800
1000
1200



5. Ajuste dos Modelos

Agora vamos realizar o ajuste dos modelos preditivos.

Vamos iniciar pelo modelo de regressão linear.

```
df_benchmark_ofc = df_dist_ofc.copy()

df_benchmark_ofc.head()
```

```
      Distrito  Area  Quartos  Garagem             Tipo  Aluguel
IPTU  \
2    Pinheiros  18.0        1        0      Apartamento   4000.0
661.0
4   Bela Vista  19.0        1        0  Studio e kitnet   4000.0
654.0
5         Brás  50.0        2        1      Apartamento   3800.0
787.0
10          Sé  63.0        2        0      Apartamento   1500.0
520.0
11          Sé  34.0        1        0      Apartamento   1000.0
406.0
```

```
    Tipo_ID  Distrito_ID   Zonas  Zonas_ID
2         2            3   Oeste         2
4         1            5  Centro         3
5         2            6  Centro         3
10        2           11  Centro         3
11        2           11  Centro         3
```

```
df_benchmark_ofc.drop(['Distrito', 'Tipo', 'Zonas'], axis=1,
inplace=True)
```

```
df_benchmark_ofc.head()
```

```
    Area  Quartos  Garagem  Aluguel   IPTU  Tipo_ID  Distrito_ID
Zonas_ID
2   18.0        1        0   4000.0  661.0        2            3
2
4   19.0        1        0   4000.0  654.0        1            5
3
5   50.0        2        1   3800.0  787.0        2            6
3
10  63.0        2        0   1500.0  520.0        2           11
3
11  34.0        1        0   1000.0  406.0        2           11
3
```

```
df_benchmark_ofc.shape
```

```
(3689, 8)
```

```
categorias_benchmark1 = ['Distrito_ID','Tipo_ID']
categorias_benchmark2 = ['Zonas_ID','Tipo_ID']
numericas_benchmark = ['Area', 'Quartos', 'Garagem']
target_benchmark = 'Aluguel'
```

Vamos separar em dois conjuntos: X_bench1 utilizará a categoria 'Distritos_ID' e X_bench2 utilizará a categoria 'Zonas_ID'.

```
X_bench1 = pd.get_dummies(df_benchmark_ofc.drop(['Zonas_ID'], axis=1),
columns=categorias_benchmark1)
X_bench2 = pd.get_dummies(df_benchmark_ofc.drop(['Distrito_ID'],
axis=1), columns=categorias_benchmark2)
```

```
X_bench1.head()
```

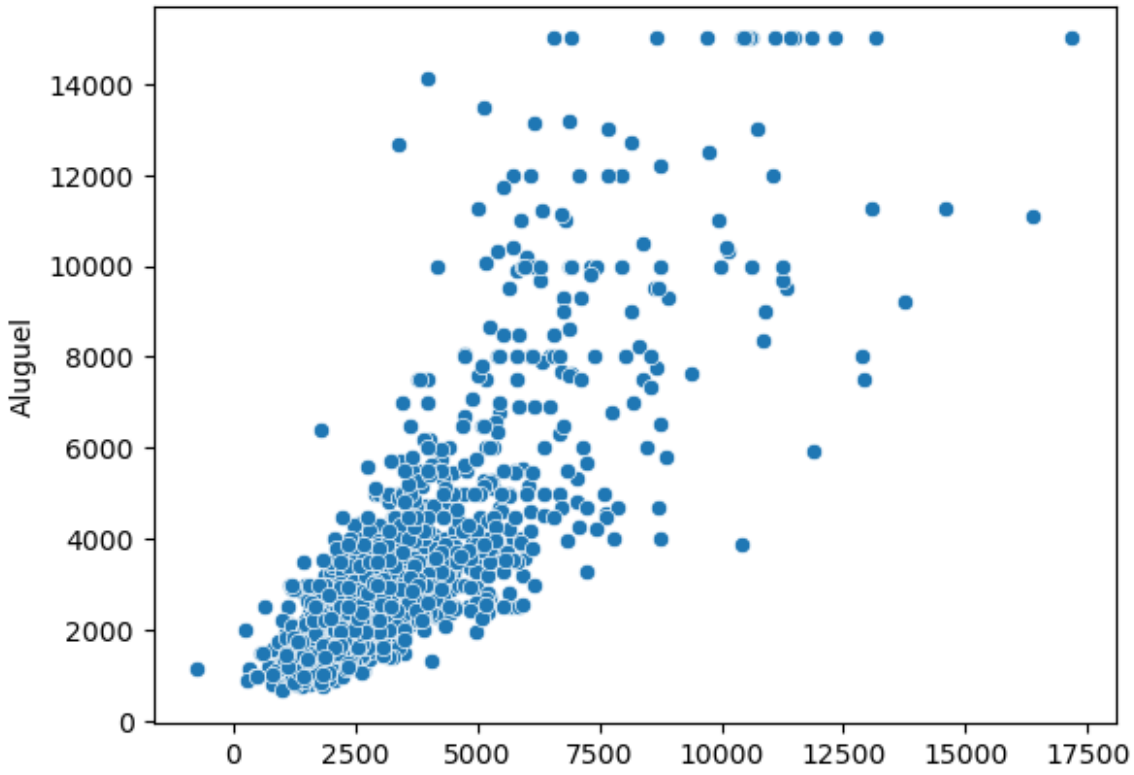
```
    Area  Quartos  Garagem  Aluguel   IPTU  Distrito_ID_3
Distrito_ID_5  \
2   18.0        1        0   4000.0  661.0           True
False
4   19.0        1        0   4000.0  654.0          False
True
```

```
5   50.0         2         1   3800.0  787.0          False
False
10  63.0         2         0   1500.0  520.0          False
False
11  34.0         1         0   1000.0  406.0          False
False

    Distrito_ID_6  Distrito_ID_11  Distrito_ID_12  ...
Distrito_ID_1063  \
2           False           False           False  ...
False
4           False           False           False  ...
False
5            True           False           False  ...
False
10          False            True           False  ...
False
11          False            True           False  ...
False

    Distrito_ID_1086  Distrito_ID_1089  Distrito_ID_1098
Distrito_ID_1122  \
2              False             False             False
False
4              False             False             False
False
5              False             False             False
False
10             False             False             False
False
11             False             False             False
False

    Distrito_ID_1129  Tipo_ID_1  Tipo_ID_2  Tipo_ID_3  Tipo_ID_4
2              False      False       True      False      False
4              False       True      False      False      False
5              False      False       True      False      False
10             False      False       True      False      False
11             False      False       True      False      False

[5 rows x 79 columns]
```

```python
def regressao_linear(df, target_column):
    X = df.drop(columns=[target_column])
    y = df[target_column]

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

    modelo = LinearRegression()
```

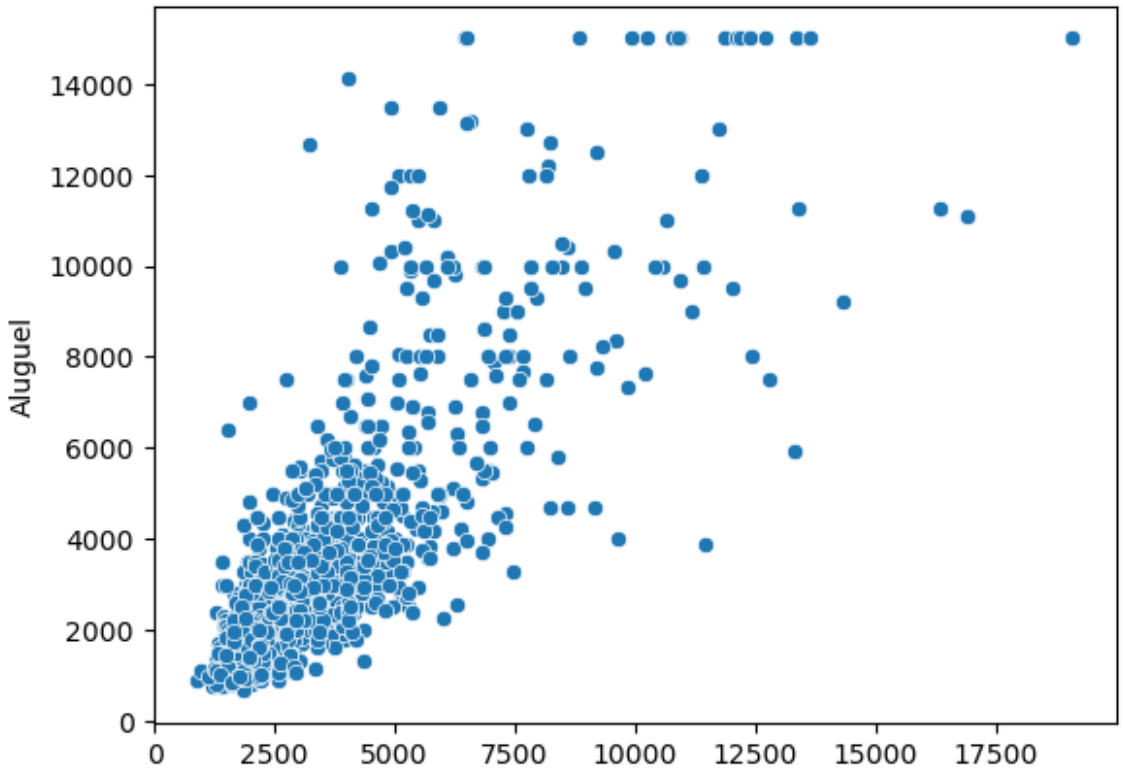
```
    modelo.fit(X_train, y_train)

    y_pred = modelo.predict(X_test)
    mse = mean_squared_error(y_test, y_pred).round(2)
    r2 = r2_score(y_test, y_pred).round(2)

    print(f'Mean Squared Error: {mse}')
    print(f'R^2 Score: {r2}')
    ax = sns.scatterplot(x=y_pred, y=y_test)
    return


regressao_linear(X_bench1, target_benchmark)
```

```
Mean Squared Error: 2496173.65
R^2 Score: 0.66
```




```
regressao_linear(X_bench2, target_benchmark)
```

```
Mean Squared Error: 2684160.43
R^2 Score: 0.63
```



Após ajustarmos o modelo de regressão linear, vamos otimizar seus hiperparametros, buscando um melhor ajuste do modelo ao nosso conjunto de dados.

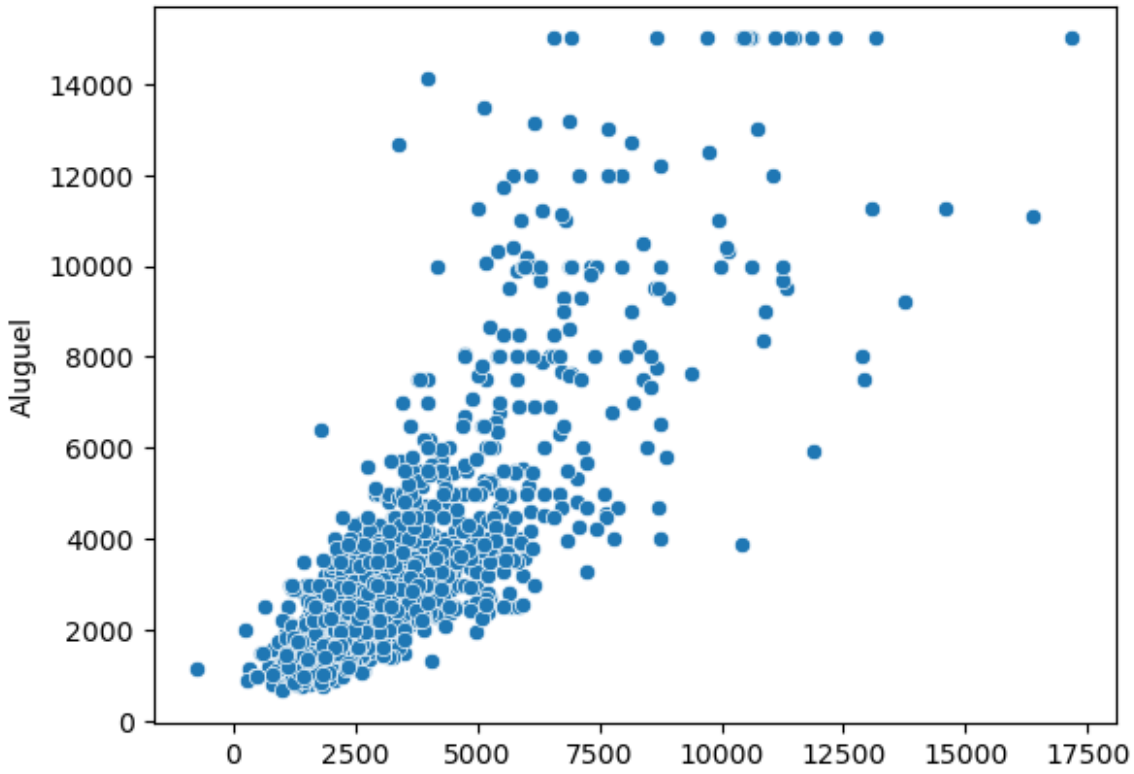
```
def otimiza_hiperparametros_regressao_linear(df, target_column):
    X = df.drop(columns=[target_column])
    y = df[target_column]

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

    pipeline = Pipeline([
        ('scaler', StandardScaler()),
        ('regressor', LinearRegression())
    ])

    param_grid = {
        'regressor__fit_intercept': [True, False],
        'regressor__copy_X': [True, False]
    }

    grid_search = GridSearchCV(pipeline, param_grid, cv=5,
scoring='neg_mean_squared_error')

    grid_search.fit(X_train, y_train)

    best_params = grid_search.best_params_
```

```
    print("Melhores hiperparâmetros:", best_params)

    best_model = grid_search.best_estimator_
    y_pred = best_model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    print("Erro médio quadrático (MSE):", mse)
    return

otimiza_hiperparametros_regressao_linear(X_bench1, target_benchmark)
```

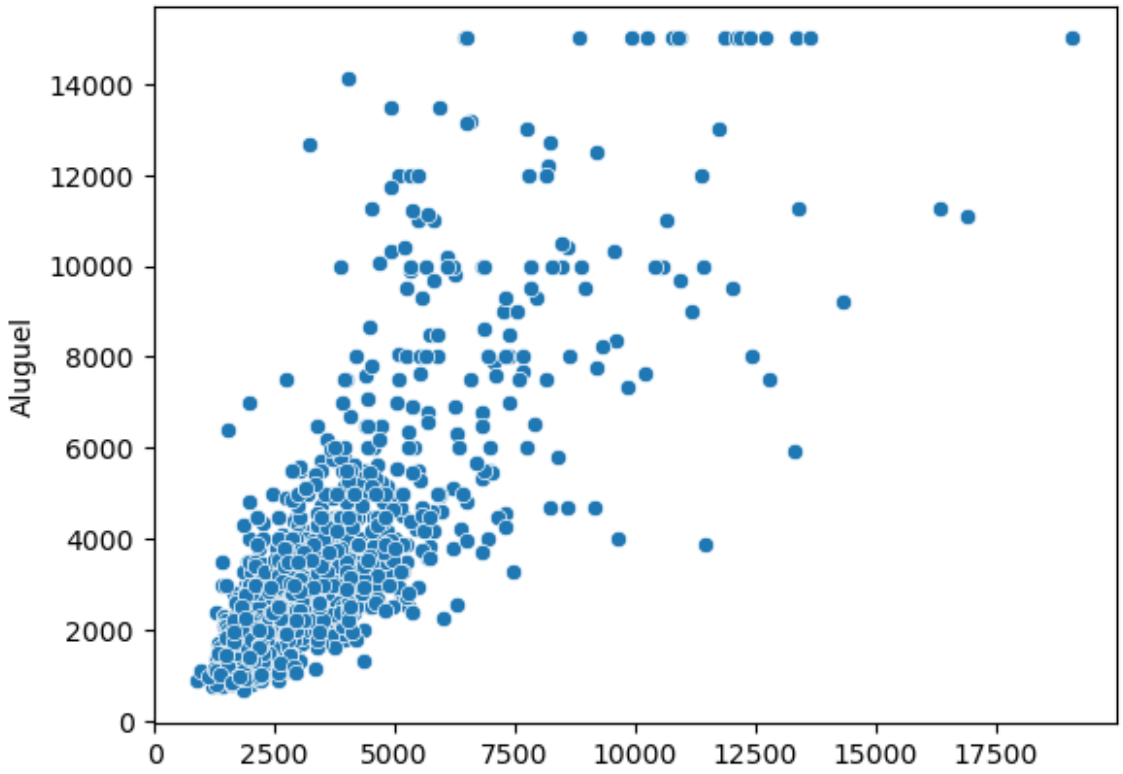
```
Melhores hiperparâmetros: {'regressor__copy_X': True,
'regressor__fit_intercept': True}
Erro médio quadrático (MSE): 2.040357871240986e+28
```

```
def regressao_linear_otimizado(df, target_column):
    X = df.drop(columns=[target_column])
    y = df[target_column]

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

    modelo = LinearRegression(copy_X=True, fit_intercept= True)

    modelo.fit(X_train, y_train)

    y_pred = modelo.predict(X_test)
    mse = mean_squared_error(y_test, y_pred).round(2)
    r2 = r2_score(y_test, y_pred).round(2)

    print(f'Mean Squared Error: {mse}')
    print(f'R^2 Score: {r2}')
    ax = sns.scatterplot(x=y_pred, y=y_test)
    return

regressao_linear_otimizado(X_bench1, target_benchmark)
```

```
Mean Squared Error: 2496173.65
R^2 Score: 0.66
```




```
regressao_linear_otimizado(X_bench2, target_benchmark)

Mean Squared Error: 2684160.43
R^2 Score: 0.63
```



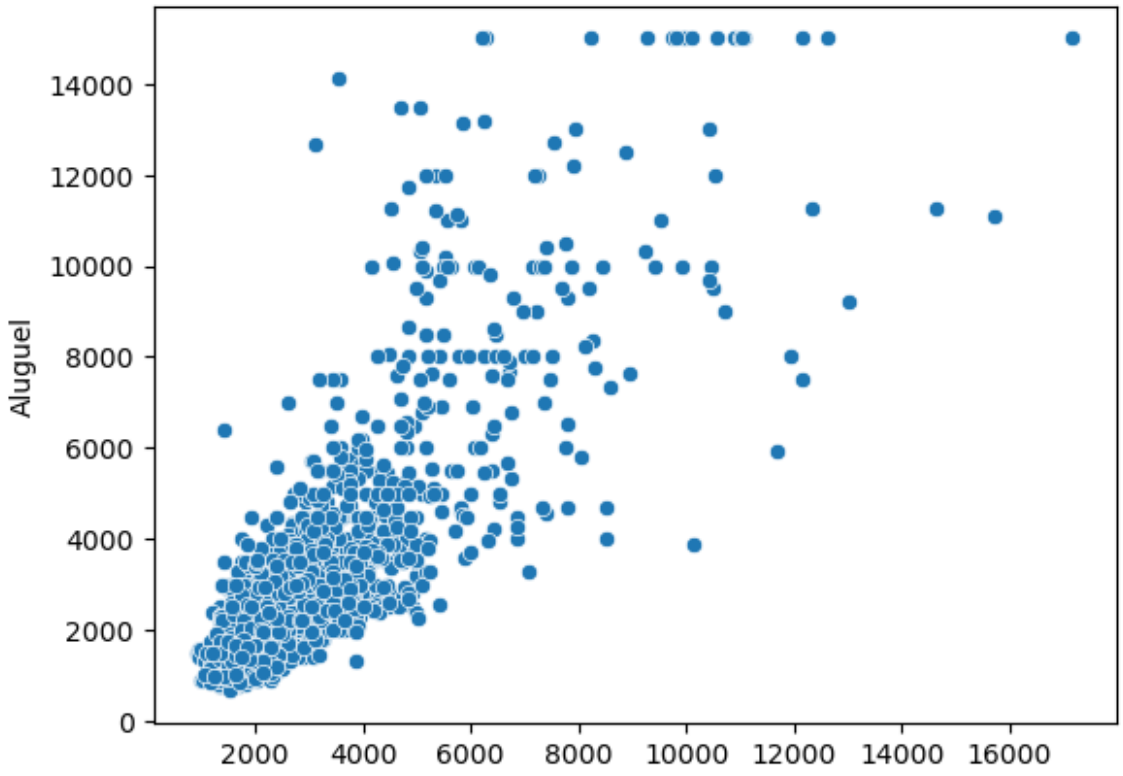
Mesmo com a otimização dos hiperparametros, não houve melhoria considerável do R^2.

Como não houve um tratamento detalhado sobre os outliers que observamos, agora vamos ajustar o modelo de regressão linear Huber.

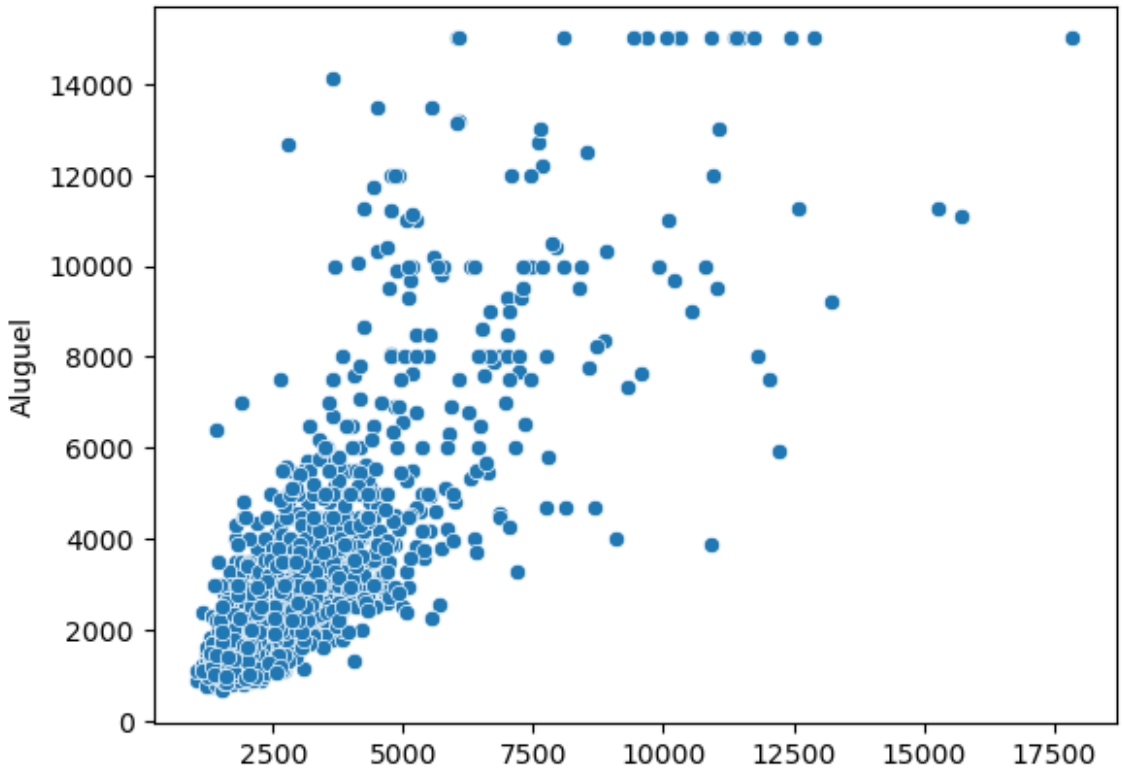
Esse modelo possui maior robustez ao lidar com outliers.

```
def regressao_linear_huber(df, target_column):
    X = df.drop(columns=[target_column])
    y = df[target_column]

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

    modelo = HuberRegressor()

    modelo.fit(X_train, y_train)

    y_pred = modelo.predict(X_test)
    mse = mean_squared_error(y_test, y_pred).round(2)
    r2 = r2_score(y_test, y_pred).round(2)

    print(f'Mean Squared Error: {mse}')
    print(f'R^2 Score: {r2}')
    ax = sns.scatterplot(x=y_pred, y=y_test)
    return
```

```
regressao_linear_huber(X_bench1, target_benchmark)

Mean Squared Error: 2645315.85
R^2 Score: 0.64

c:\Users\User\anaconda3\lib\site-packages\sklearn\linear_model\
_huber.py:342: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
  self.n_iter_ = _check_optimize_result("lbfgs", opt_res,
self.max_iter)
```




```
regressao_linear_huber(X_bench2, target_benchmark)

Mean Squared Error: 2789746.08
R^2 Score: 0.62
```



Também vamos otimizar seus hiperparametros.

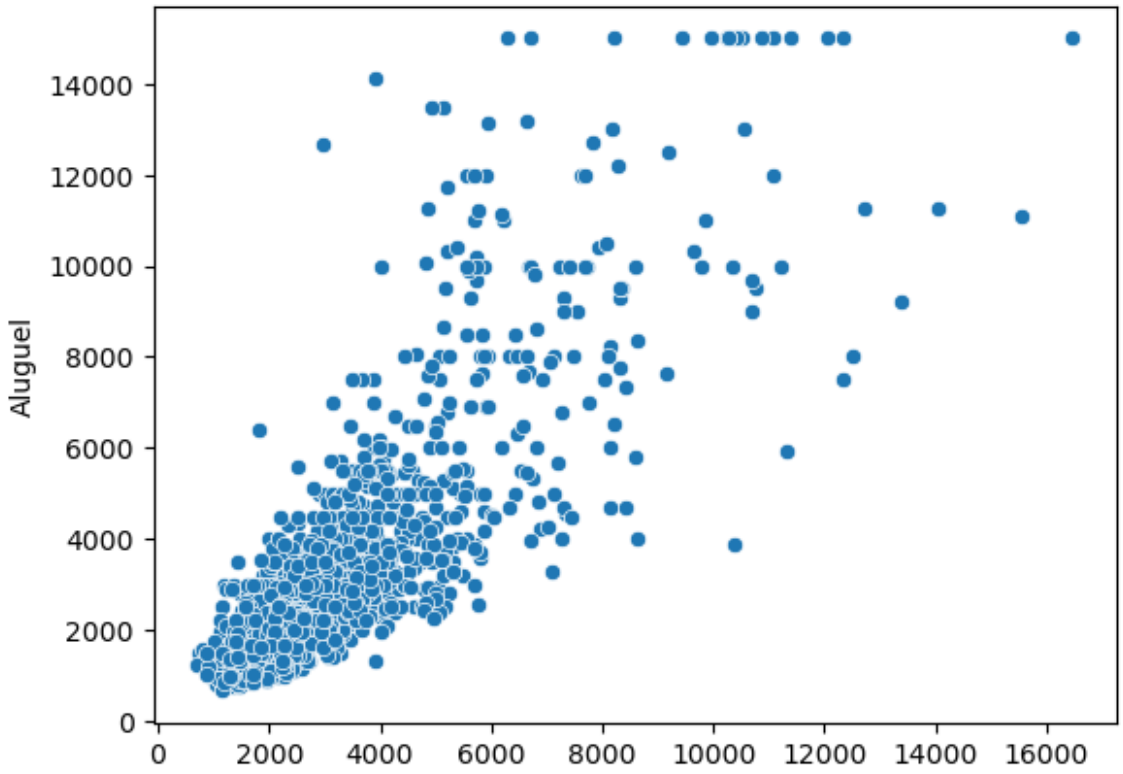
```
def otimiza_hiperparametros_regressao_linear_huber(df, target_column):
    X = df.drop(columns=[target_column])
    y = df[target_column]

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

    pipeline = Pipeline([
        ('scaler', StandardScaler()),
        ('regressor', HuberRegressor())
    ])

    param_grid = {
        'regressor__epsilon': [1.1, 1.35, 1.5, 1.75, 2.0],
        'regressor__alpha': [0.0001, 0.001, 0.01, 0.1, 1.0],
        'regressor__max_iter': [300, 400, 500, 600],
        'regressor__tol': [1e-3, 1e-4, 1e-5]
    }

    grid_search = GridSearchCV(pipeline, param_grid, cv=5,
scoring='neg_mean_squared_error')

    grid_search.fit(X_train, y_train)
```

```
    best_params = grid_search.best_params_
    print("Melhores hiperparâmetros:", best_params)

    best_model = grid_search.best_estimator_
    y_pred = best_model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    print("Erro médio quadrático (MSE):", mse)
    return

otimiza_hiperparametros_regressao_linear_huber(X_bench1,
target_benchmark)
```

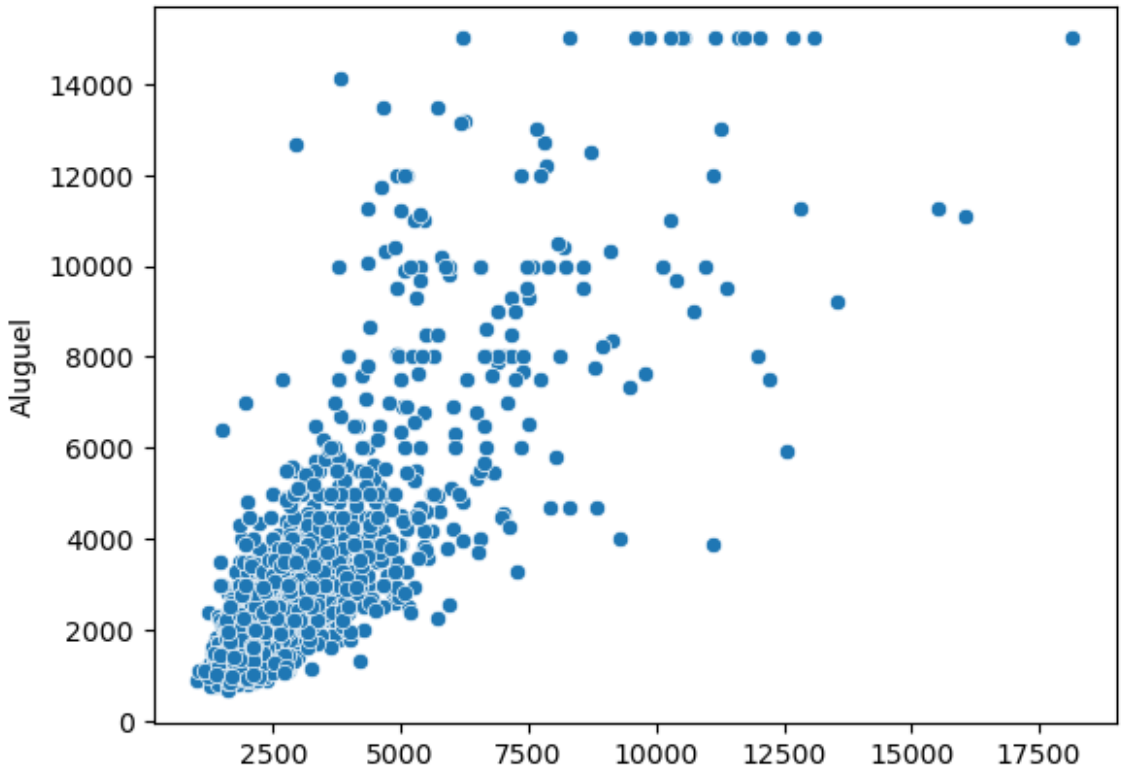
```
Melhores hiperparâmetros: {'regressor__alpha': 0.0001,
'regressor__epsilon': 2.0, 'regressor__max_iter': 300,
'regressor__tol': 0.001}
Erro médio quadrático (MSE): 2546307.832250407
```

```
def regressao_linear_huber_otimizado(df, target_column):
    X = df.drop(columns=[target_column])
    y = df[target_column]

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

    modelo = HuberRegressor(alpha=0.0001, epsilon=2.0, max_iter=400,
tol=0.001)

    modelo.fit(X_train, y_train)

    y_pred = modelo.predict(X_test)
    mse = mean_squared_error(y_test, y_pred).round(2)
    r2 = r2_score(y_test, y_pred).round(2)

    print(f'Mean Squared Error: {mse}')
    print(f'R^2 Score: {r2}')
    ax = sns.scatterplot(x=y_pred, y=y_test)
    return

regressao_linear_huber_otimizado(X_bench1, target_benchmark)
```

```
Mean Squared Error: 2523173.49
R^2 Score: 0.65
```




```
regressao_linear_huber_otimizado(X_bench2, target_benchmark)

Mean Squared Error: 2726664.7
R^2 Score: 0.63
```



Otimização que também não impacto considerável no ajuste.

Como nenhum dos modelos de regressão linear se mostraram com alta eficiência, vamos testar ajustar dois modelos extras.

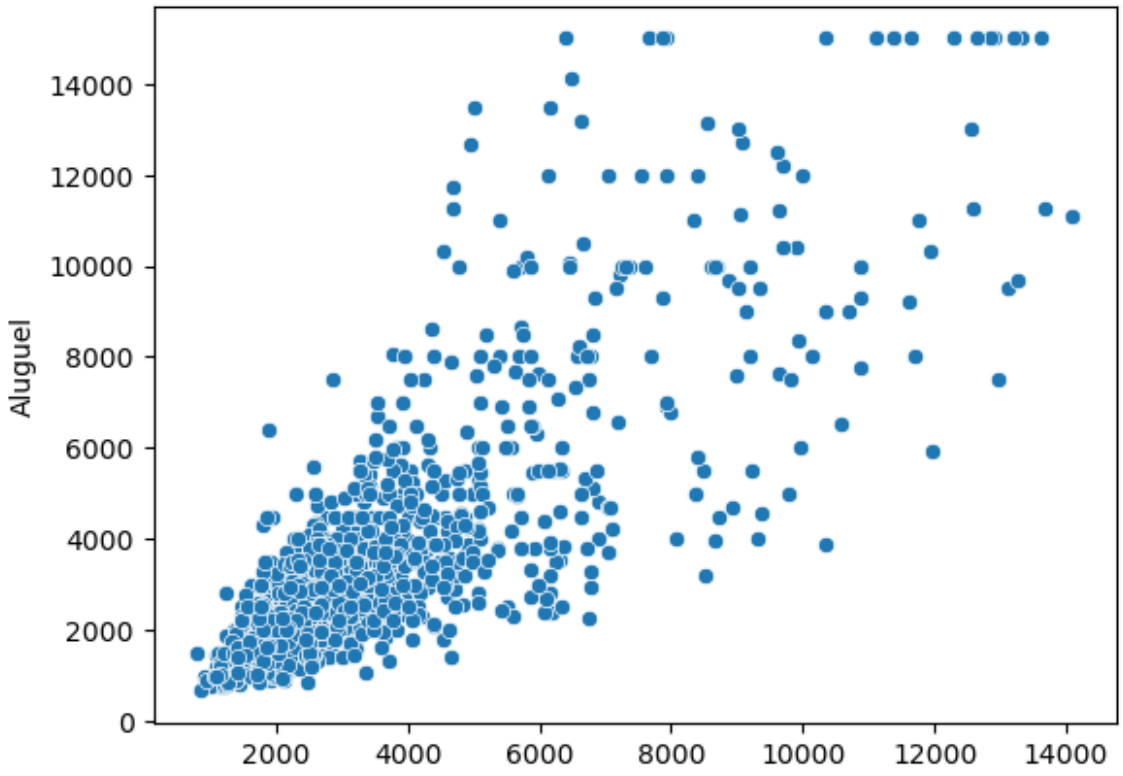
Iremos ajustar o modelo Random Forest e o Gradient Boosting Tree.

Esses modelos foram escolhidos por tenderem a ser mais precisos que os modelos de regressão linear, mas com processamento mais lento.

```
def random_forest(df, target_column):
    X = df.drop(columns=[target_column])
    y = df[target_column]

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

    modelo = RandomForestRegressor(n_estimators=100, random_state=42)

    modelo.fit(X_train, y_train)

    y_pred = modelo.predict(X_test)
    mse = mean_squared_error(y_test, y_pred).round(2)
    r2 = r2_score(y_test, y_pred).round(2)

    print(f'Mean Squared Error: {mse}')
```

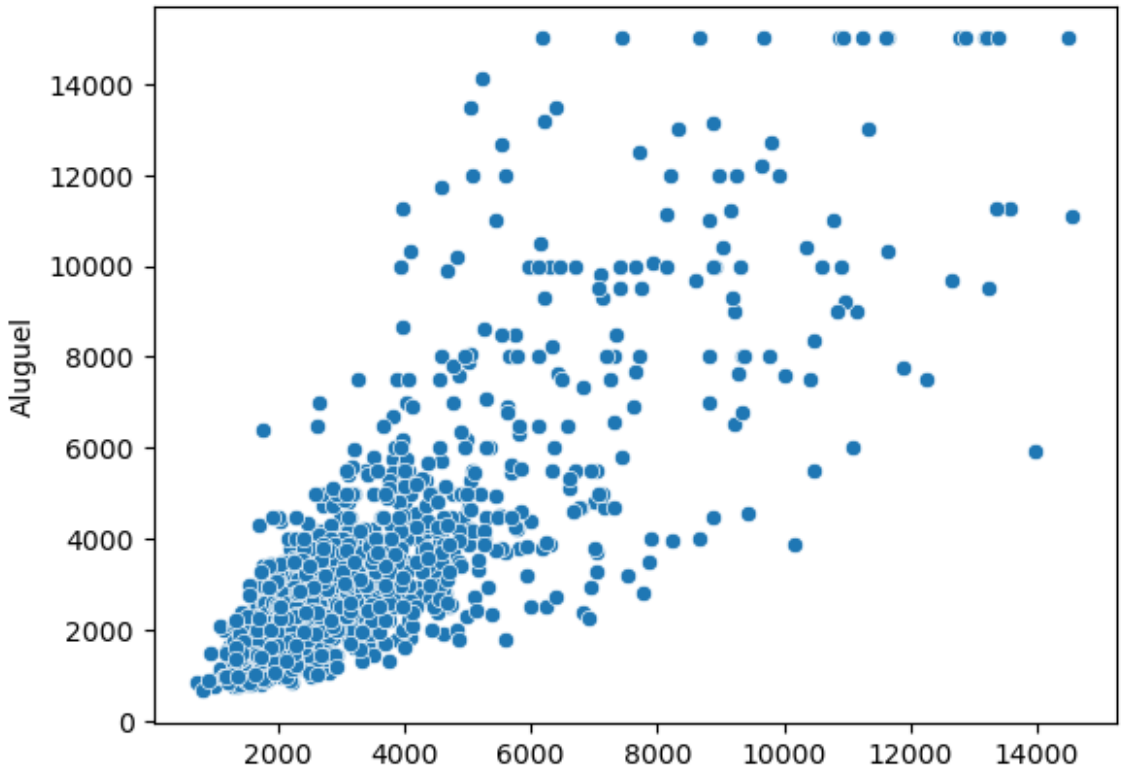
```
    print(f'R^2 Score: {r2}')
    ax = sns.scatterplot(x=y_pred, y=y_test)
    return

random_forest(X_bench1, target_benchmark)
```

```
Mean Squared Error: 2438033.41
R^2 Score: 0.67
```




```
random_forest(X_bench2, target_benchmark)
```

```
Mean Squared Error: 2569330.45
R^2 Score: 0.65
```



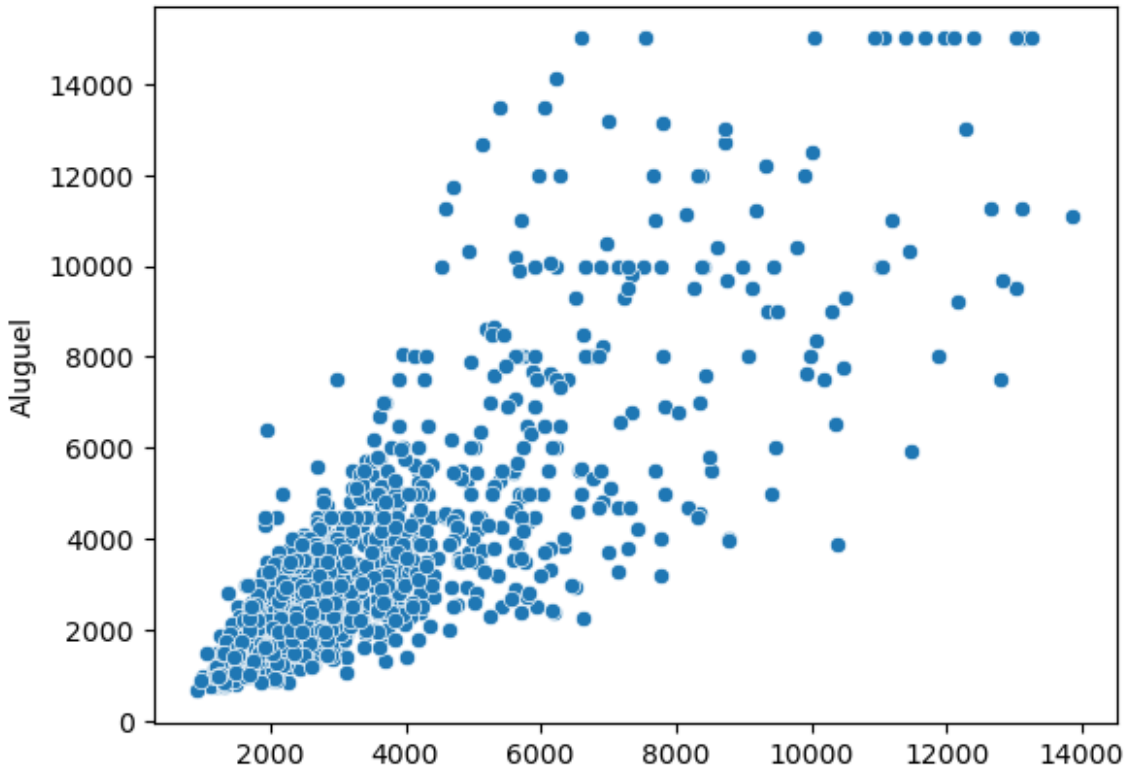

```
def otimiza_hiperparametros_random_forest(df, target_column):
    X = df.drop(columns=[target_column])
    y = df[target_column]

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

    pipeline = Pipeline([
        ('scaler', StandardScaler()),
        ('regressor', RandomForestRegressor())
    ])

    param_grid = {
        'regressor__n_estimators': [100, 200, 300],
        'regressor__max_depth': [None, 10, 20],
        'regressor__min_samples_split': [2, 5, 10]
    }

    grid_search = GridSearchCV(pipeline, param_grid, cv=5,
scoring='neg_mean_squared_error')

    grid_search.fit(X_train, y_train)

    best_params = grid_search.best_params_
    print("Melhores hiperparâmetros:", best_params)
```

```
    best_model = grid_search.best_estimator_
    y_pred = best_model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    print("Erro médio quadrático (MSE):", mse)
    return

otimiza_hiperparametros_random_forest(X_bench1, target_benchmark)
```

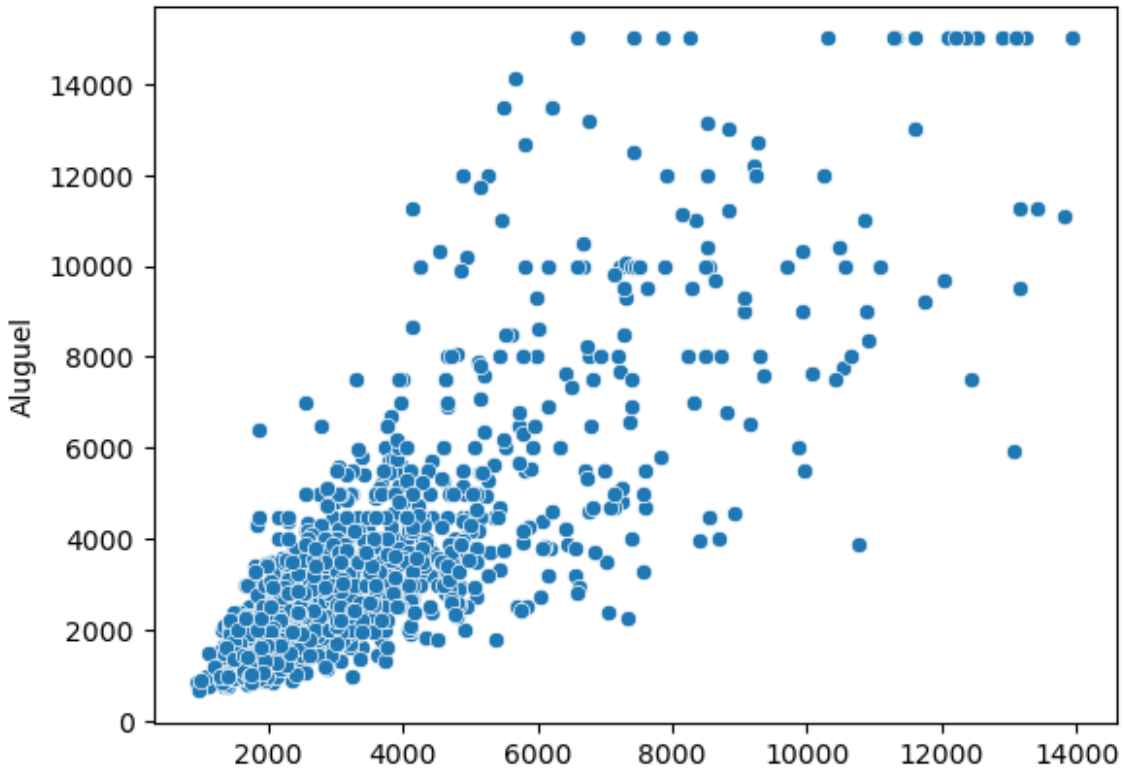
```
Melhores hiperparâmetros: {'regressor__max_depth': 20,
'regressor__min_samples_split': 10, 'regressor__n_estimators': 300}
Erro médio quadrático (MSE): 2381852.0783634656
```

```
def random_forest_otimizado(df, target_column):
    X = df.drop(columns=[target_column])
    y = df[target_column]

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

    modelo = RandomForestRegressor(max_depth=20, min_samples_split=10,
n_estimators=300, random_state=42)

    modelo.fit(X_train, y_train)

    y_pred = modelo.predict(X_test)
    mse = mean_squared_error(y_test, y_pred).round(2)
    r2 = r2_score(y_test, y_pred).round(2)

    print(f'Mean Squared Error: {mse}')
    print(f'R^2 Score: {r2}')
    ax = sns.scatterplot(x=y_pred, y=y_test)
    return

random_forest_otimizado(X_bench1, target_benchmark)
```

```
Mean Squared Error: 2381659.33
R^2 Score: 0.67
```

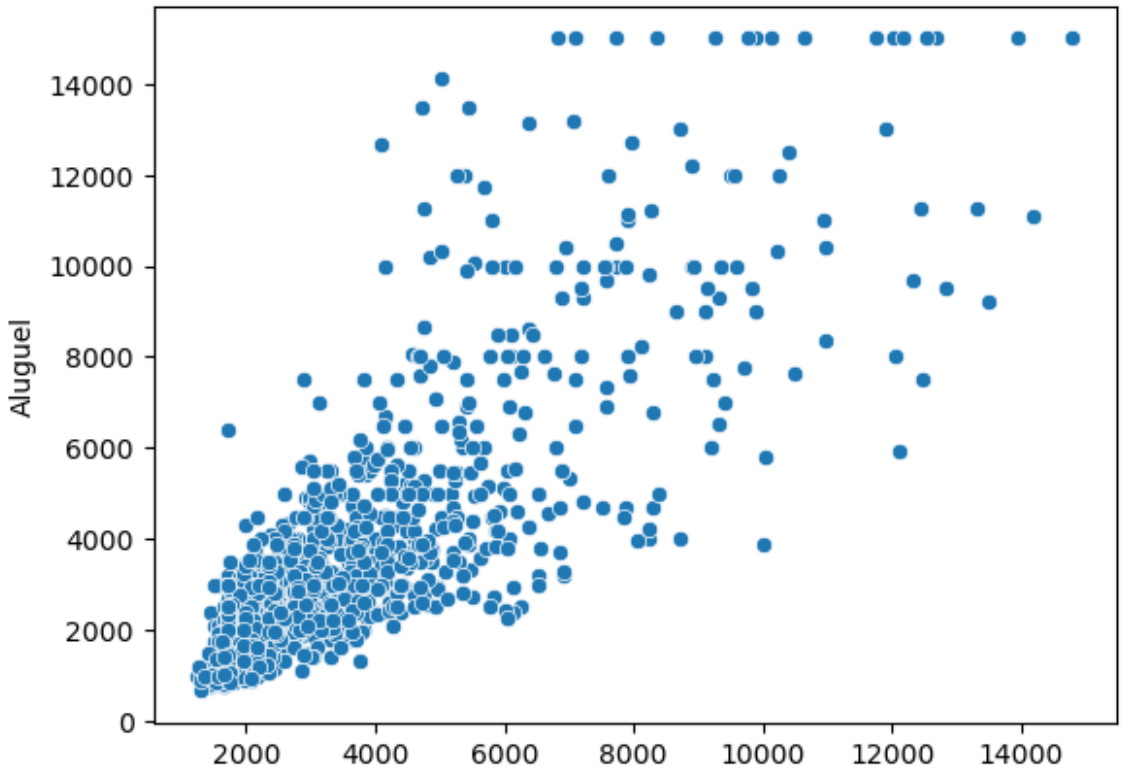



```
random_forest_otimizado(X_bench2, target_benchmark)

Mean Squared Error: 2441240.06
R^2 Score: 0.67
```

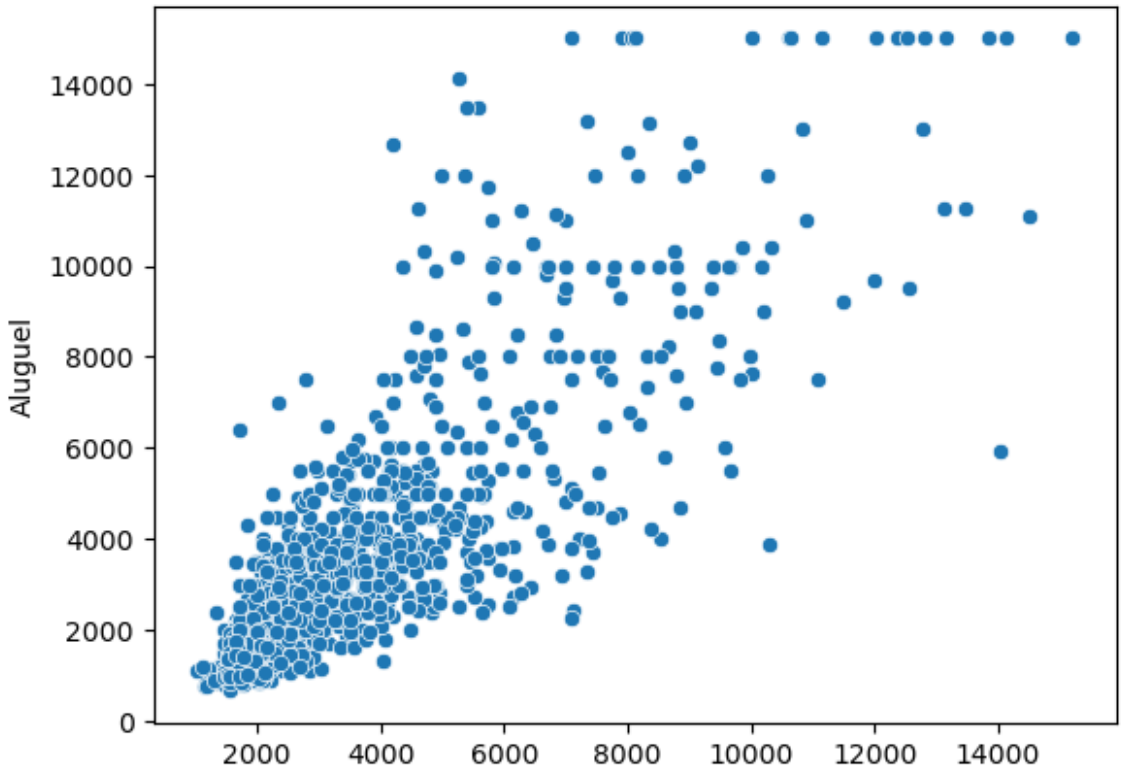



```
def gradient_boosting_tree(df, target_column):
    X = df.drop(columns=[target_column])
    y = df[target_column]

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

    modelo = GradientBoostingRegressor(n_estimators=100,
learning_rate=0.1, max_depth=3, random_state=42)

    modelo.fit(X_train, y_train)

    y_pred = modelo.predict(X_test)
    mse = mean_squared_error(y_test, y_pred).round(2)
    r2 = r2_score(y_test, y_pred).round(2)

    print(f'Mean Squared Error: {mse}')
    print(f'R^2 Score: {r2}')
    ax = sns.scatterplot(x=y_pred, y=y_test)
    return

gradient_boosting_tree(X_bench1, target_benchmark)
```

```
Mean Squared Error: 2443607.24
R^2 Score: 0.67
```




```
gradient_boosting_tree(X_bench2, target_benchmark)

Mean Squared Error: 2429261.39
R^2 Score: 0.67
```



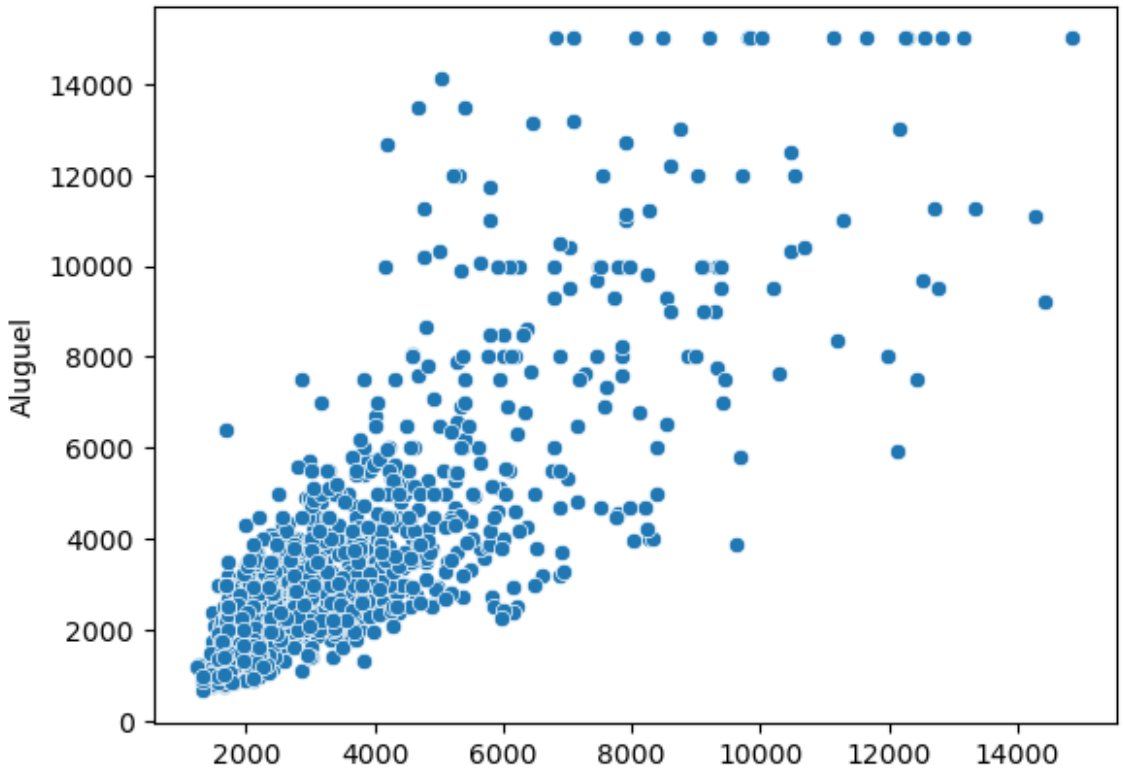

```
def otimiza_hiperparametros_gbt(df, target_column):
    X = df.drop(columns=[target_column])
    y = df[target_column]

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

    pipeline = Pipeline([
        ('scaler', StandardScaler()),
        ('regressor', GradientBoostingRegressor())
    ])

    param_grid = {
        'regressor__n_estimators': [100, 200, 300],
        'regressor__max_depth': [3, 5, 7],
        'regressor__learning_rate': [0.01, 0.1, 0.5],
        'regressor__min_samples_split': [2, 5, 10]
    }

    grid_search = GridSearchCV(pipeline, param_grid, cv=5,
scoring='neg_mean_squared_error')

    grid_search.fit(X_train, y_train)

    best_params = grid_search.best_params_
    print("Melhores hiperparâmetros:", best_params)
```

```
    best_model = grid_search.best_estimator_
    y_pred = best_model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    print("Erro médio quadrático (MSE):", mse)
    return

otimiza_hiperparametros_gbt(X_bench1, target_benchmark)
```

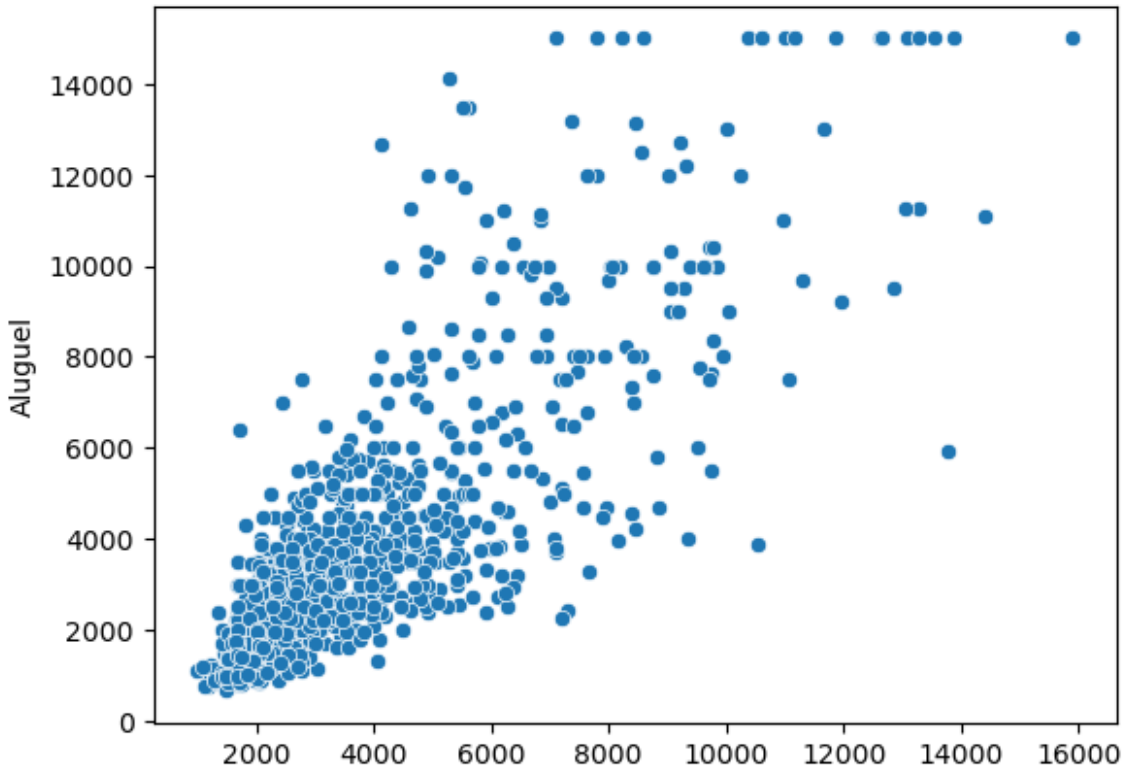
```
Melhores hiperparâmetros: {'regressor__learning_rate': 0.1,
'regressor__max_depth': 3, 'regressor__min_samples_split': 10,
'regressor__n_estimators': 100}
Erro médio quadrático (MSE): 2434350.8974268353
```

```
def gradient_boosting_tree_otimizado(df, target_column):
    X = df.drop(columns=[target_column])
    y = df[target_column]

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

    modelo = GradientBoostingRegressor(n_estimators=100,
learning_rate=0.1, max_depth=3,min_samples_split=10, random_state=42)

    modelo.fit(X_train, y_train)

    y_pred = modelo.predict(X_test)
    mse = mean_squared_error(y_test, y_pred).round(2)
    r2 = r2_score(y_test, y_pred).round(2)

    print(f'Mean Squared Error: {mse}')
    print(f'R^2 Score: {r2}')
    ax = sns.scatterplot(x=y_pred, y=y_test)
    return

gradient_boosting_tree_otimizado(X_bench1, target_benchmark)
```

```
Mean Squared Error: 2434851.31
R^2 Score: 0.67
```




```
gradient_boosting_tree_otimizado(X_bench2, target_benchmark)

Mean Squared Error: 2426227.84
R^2 Score: 0.67
```



6. Conclusão

Neste projeto realizamos a análise de um conjunto de dados de preços de aluguéis em São Paulo utilizando Python e algumas de suas bibliotecas. O projeto abrangiu a importação, tratamento, visualização e ajuste de modelos preditivos.

Assim que iniciamos o tratamento dos dados, observamos que a distribuição geográfica das informações estava bastante complexa, com 1199 distritos. Com esse valor de distritos, sugerimos a hipotese de que o conjunto de dados é sobre a região metropolitana de São Paulo. Dessa forma, com essa distribuição seria muito dificil realizar o ajuste de um modelo de regressão linear, pois é um modelo bastante simples, então o conjunto foi divido entre os distritos oficiais da cidade de São Paulo (também adicionamos a qual zona da cidade cada distrito pertence) e os outros. Após realizarmos esses agrupamentos, o conjunto que decidimos dar foco (distritos oficiais) ficou com uma quantidade pequena de dados, com 3689 linhas (conjunto inicial de 11657 linhas).

Com o tratamento e preparação dos dados realizado, partimos para o ajuste dos modelos preditivos. Foram realizados ajustes para o conjunto de dados utilizando como categoria o seu distrito ou a sua zona, mas não foi observado nenhuma diferença considerável entre a escolha de categoria. Também não foi observada grande diferença da eficiência e precisão entre os modelos, uma hipótese para o resultado do R^2 (que ficou por volta de 0.6) é a quantidade de dados fornecidas para os modelos, talvez com um tratamento mais complexo para o conjunto de dados "bruto", obteriamos mais pontos e seria possível obter um resultado melhor.

De qualquer forma, com o conjunto apresentado foi possível obter valiosas informações sobre a relação das variáveis (área, quartos, garagem, aluguel) e o tipo de imóvel a partir da aplicação de técnicas de estatística, apresentando essas informações na forma de gráficos. Também iniciamos o desenvolvimento de modelos preditivos que provavelmente com mais alguns ajustes (desde o conjunto de dados até os modelos finais) seja possível ser concluido e aplicável.