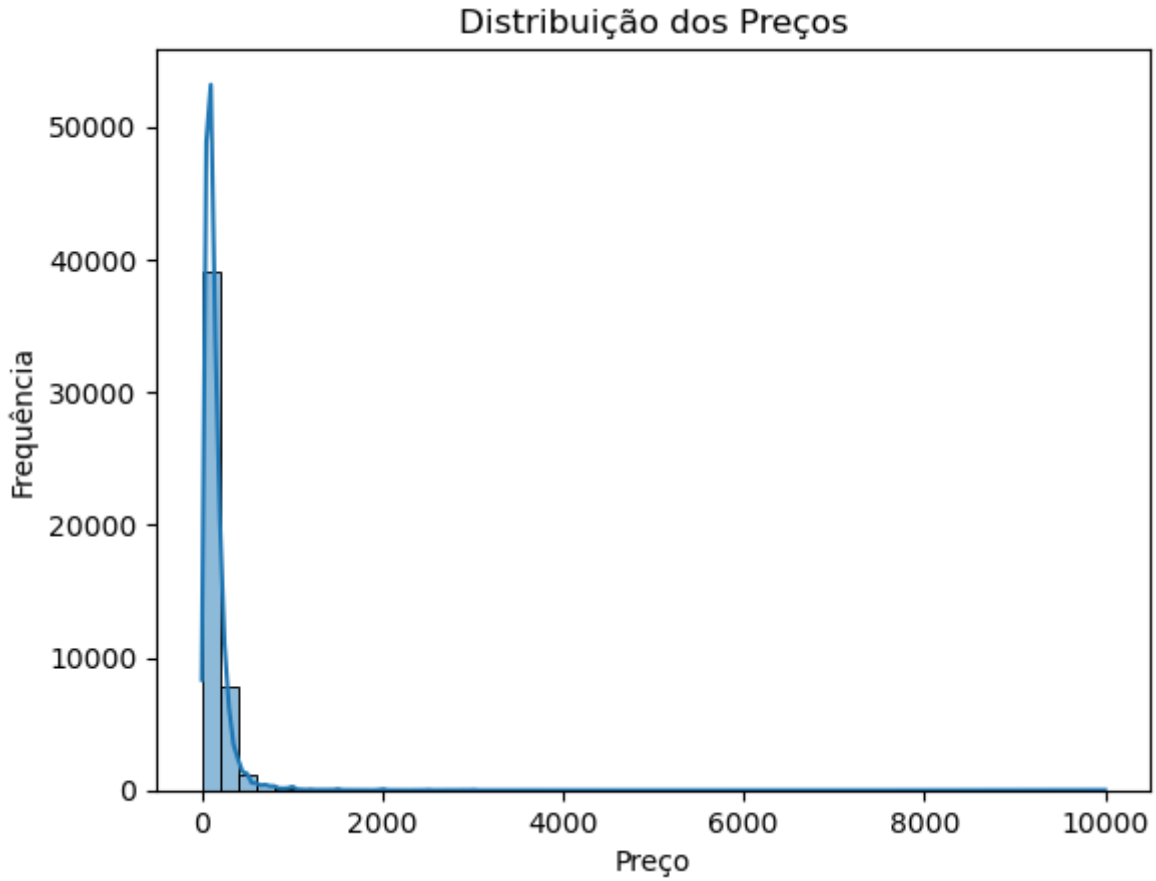
In [6]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
import joblib

# Carregar dados
data = pd.read_csv('C://Users//elisa//OneDrive//Área de Trabalho//teste_indicium_pr

# EDA simplificada
# Exemplo: Distribuição dos preços
sns.histplot(data['price'], bins=50, kde=True)
plt.xlabel('Preço')
plt.ylabel('Frequência')
plt.title('Distribuição dos Preços')
plt.show()

# Tratamento de valores ausentes
data['reviews_por_mes'] = data['reviews_por_mes'].fillna(0)

# Preparação dos dados
features = ['latitude', 'longitude', 'minimo_noites', 'numero_de_reviews', 'reviews
X = data[features]
y = data['price']

# Divisão em conjuntos de treino e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Treinamento do modelo
model = LinearRegression()
model.fit(X_train, y_train)

# Avaliação do modelo
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f'MSE: {mse}')

# Salvando o modelo
joblib.dump(model, 'modelo_preco.pkl')

# Para carregar o modelo
# model = joblib.load('modelo_preco.pkl')
```

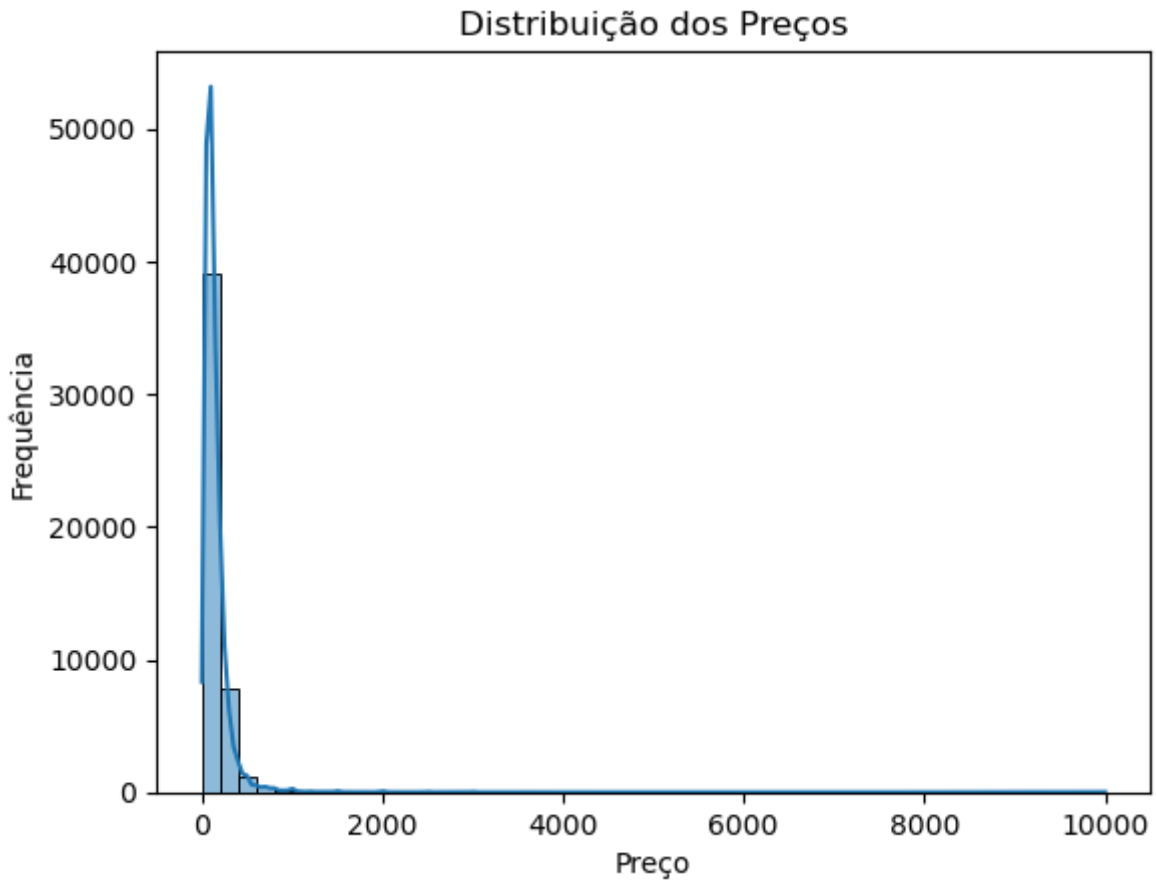



```
MSE: 48169.196534798495
```

Out[6]:
```
['modelo_preco.pkl']
```

In [7]:
```
data = pd.read_csv(r'C://Users//elisa//OneDrive//Área de Trabalho//teste_indicium_p
```

In [8]:
```
# EDA simplificada
# Exemplo: Distribuição dos preços
sns.histplot(data['price'], bins=50, kde=True)
plt.xlabel('Preço')
plt.ylabel('Frequência')
plt.title('Distribuição dos Preços')
plt.show()
```



In [10]:
```
# Tratamento de valores ausentes
data['reviews_por_mes'] = data['reviews_por_mes'].fillna(0)
```

In [11]:
```
# Preparação dos dados
features = ['latitude', 'longitude', 'minimo_noites', 'numero_de_reviews', 'reviews
X = data[features]
y = data['price']
```

In [12]:
```
# Divisão em conjuntos de treino e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

In [13]:
```
# Treinamento do modelo
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[13]:
```
▾ LinearRegression
LinearRegression()
```

In [14]:
```
# Avaliação do modelo
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f'MSE: {mse}')
```

```
MSE: 48169.196534798495
```

In [15]:
```
# Salvando o modelo
joblib.dump(model, 'modelo_preco.pkl')
```

Out[15]:
```
['modelo_preco.pkl']
```

In [16]:
```
# Importação das bibliotecas necessárias
import pandas as pd
```

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
import joblib
```

In [18]:
```
# Carregar os dados
# Substitua 'seu_caminho_aqui' pelo caminho do seu arquivo
data = pd.read_csv('C://Users//elisa//OneDrive//Área de Trabalho//teste_indicium_pr
```

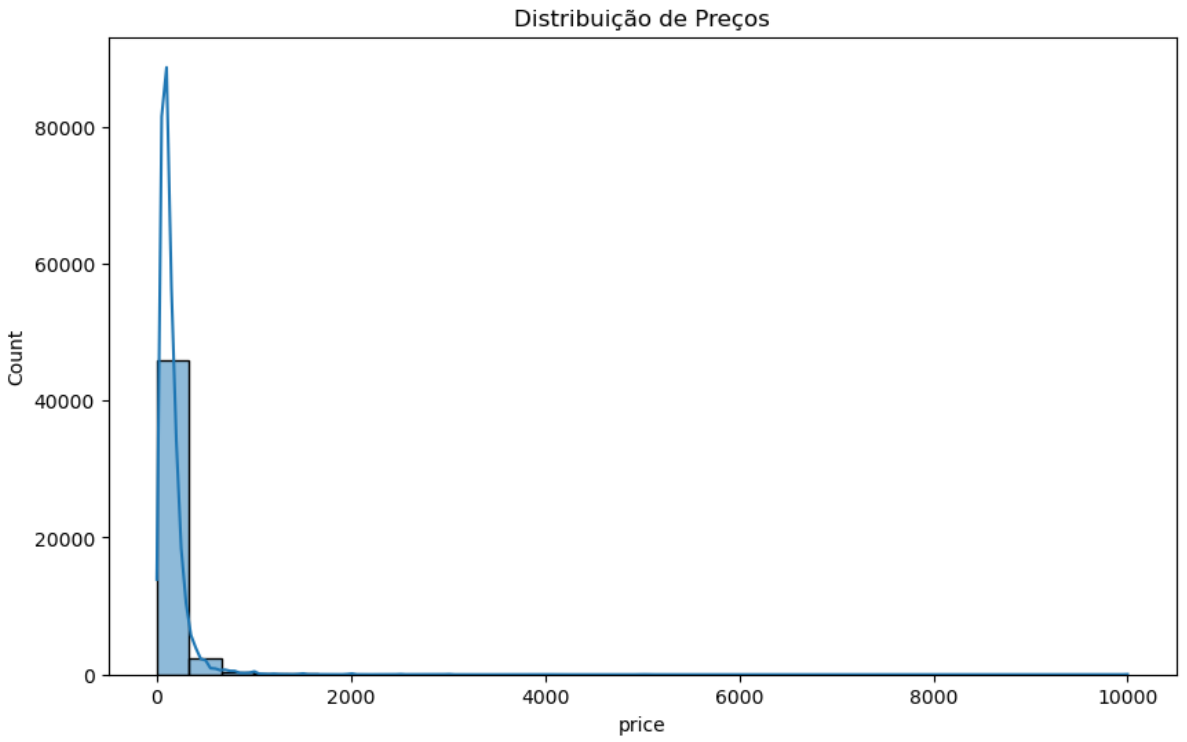
In [19]:
```
# EDA básica
print(data.head())
print(data.describe())
```

```
     id                                              nome  host_id  \
0  2595                              Skylit Midtown Castle     2845
1  3647                THE VILLAGE OF HARLEM....NEW YORK !     4632
2  3831                    Cozy Entire Floor of Brownstone     4869
3  5022  Entire Apt: Spacious Studio/Loft by central park     7192
4  5099         Large Cozy 1 BR Apartment In Midtown East     7322

     host_name bairro_group        bairro  latitude  longitude  \
0     Jennifer    Manhattan       Midtown  40.75362  -73.98377
1   Elisabeth    Manhattan        Harlem  40.80902  -73.94190
2  LisaRoxanne     Brooklyn  Clinton Hill  40.68514  -73.95976
3        Laura    Manhattan   East Harlem  40.79851  -73.94399
4        Chris    Manhattan   Murray Hill  40.74767  -73.97500

         room_type  price  minimo_noites  numero_de_reviews ultima_review  \
0  Entire home/apt    225              1                 45    2019-05-21
1     Private room    150              3                  0           NaN
2  Entire home/apt     89              1                270    2019-07-05
3  Entire home/apt     80             10                  9    2018-11-19
4  Entire home/apt    200              3                 74    2019-06-22

   reviews_por_mes  calculado_host_listings_count  disponibilidade_365
0             0.38                              2                  355
1              NaN                              1                  365
2             4.64                              1                  194
3             0.10                              1                    0
4             0.59                              1                  129
                 id       host_id      latitude     longitude         price  \
count  4.889400e+04  4.889400e+04  48894.000000  48894.000000  48894.000000
mean   1.901753e+07  6.762139e+07     40.728951    -73.952169    152.720763
std    1.098288e+07  7.861118e+07      0.054529      0.046157    240.156625
min    2.595000e+03  2.438000e+03     40.499790    -74.244420      0.000000
25%    9.472371e+06  7.822737e+06     40.690100    -73.983070     69.000000
50%    1.967743e+07  3.079553e+07     40.723075    -73.955680    106.000000
75%    2.915225e+07  1.074344e+08     40.763117    -73.936273    175.000000
max    3.648724e+07  2.743213e+08     40.913060    -73.712990  10000.000000

       minimo_noites  numero_de_reviews  reviews_por_mes  \
count   48894.000000       48894.000000     38842.000000
mean        7.030085          23.274758         1.373251
std        20.510741          44.550991         1.680453
min         1.000000           0.000000         0.010000
25%         1.000000           1.000000         0.190000
50%         3.000000           5.000000         0.720000
75%         5.000000          24.000000         2.020000
max      1250.000000         629.000000        58.500000

       calculado_host_listings_count  disponibilidade_365
count                   48894.000000         48894.000000
mean                        7.144005           112.776169
std                        32.952855           131.618692
min                         1.000000             0.000000
25%                         1.000000             0.000000
50%                         1.000000            45.000000
75%                         2.000000           227.000000
max                       327.000000           365.000000
```

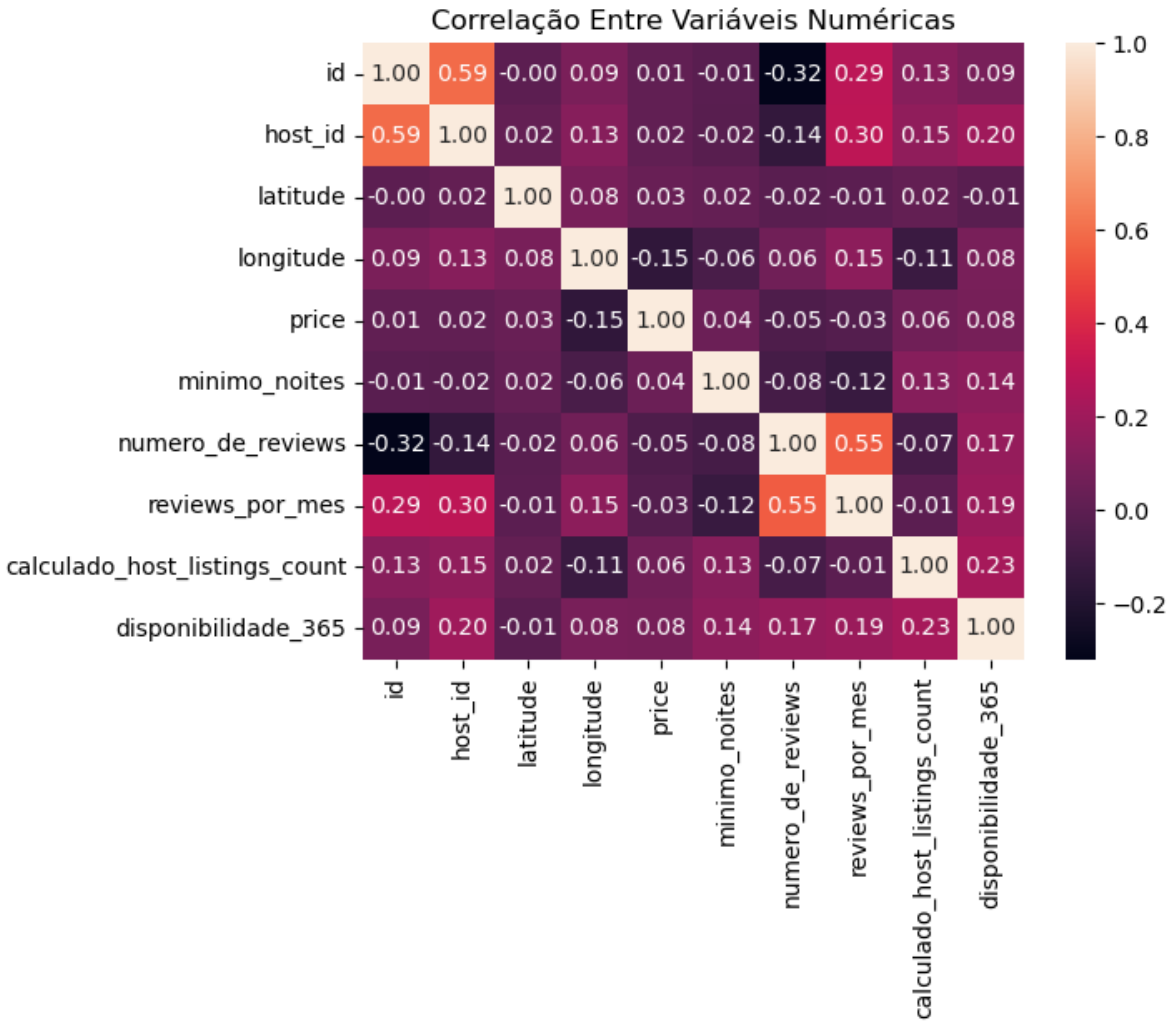
In [20]:
```
# Verificar valores ausentes
print(data.isnull().sum())
```

```
id                                  0
nome                               16
host_id                             0
host_name                          21
bairro_group                        0
bairro                              0
latitude                            0
longitude                           0
room_type                           0
price                               0
minimo_noites                       0
numero_de_reviews                   0
ultima_review                   10052
reviews_por_mes                 10052
calculado_host_listings_count       0
disponibilidade_365                 0
dtype: int64
```

In [21]:
```
# Visualização da distribuição de preços
plt.figure(figsize=(10, 6))
sns.histplot(data['price'], bins=30, kde=True)
plt.title('Distribuição de Preços')
plt.show()
```



In [23]:
```
# Selecionar apenas colunas numéricas para calcular a correlação
corr_data = data.select_dtypes(include=[np.number])
sns.heatmap(corr_data.corr(), annot=True, fmt='.2f')
plt.title('Correlação Entre Variáveis Numéricas')
plt.show()
```



In [24]:
```
# Pré-processamento e preparação dos dados
# Aqui você pode adicionar ou remover etapas conforme necessário

# Tratamento de valores ausentes
# Exemplo: Preenchendo valores ausentes com a média ou mediana
data['reviews_por_mes'] = data['reviews_por_mes'].fillna(data['reviews_por_mes'].me
```

In [25]:
```
# Divisão dos dados em características e alvo
X = data[['latitude', 'longitude', 'minimo_noites', 'numero_de_reviews', 'reviews_p
y = data['price']
```

In [26]:
```
# Divisão dos dados em características e alvo
X = data[['latitude', 'longitude', 'minimo_noites', 'numero_de_reviews', 'reviews_p
y = data['price']
```

In [27]:
```
# Divisão dos dados em treino e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

In [28]:
```
# Construção do modelo
# Utilizando um modelo linear como exemplo

# Pipeline de pré-processamento para transformações numéricas
numeric_features = ['latitude', 'longitude', 'minimo_noites', 'numero_de_reviews',
numeric_transformer = Pipeline(steps=[
    ('scaler', StandardScaler())
])
```

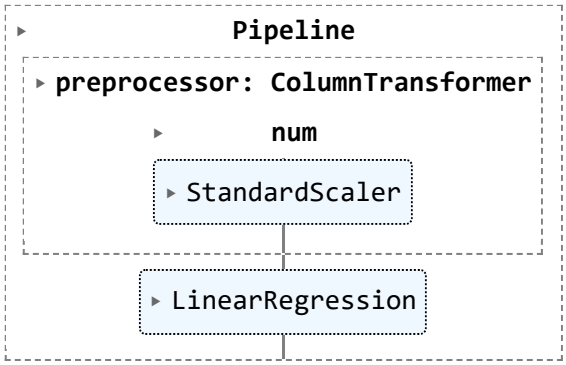
In [29]:
```
# Combinando transformações numéricas
preprocessor = ColumnTransformer(
```

```
    transformers=[
        ('num', numeric_transformer, numeric_features),
    ])
```

In [30]:
```
# Pipeline do modelo
model = Pipeline(steps=[('preprocessor', preprocessor),
                        ('classifier', LinearRegression())])
```

In [31]:
```
# Treinamento do modelo
model.fit(X_train, y_train)
```

Out[31]:



In [32]:
```
# Avaliação do modelo
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f'MSE: {mse}')
```

```
MSE: 48171.08164548659
```

In [34]:
```
from joblib import dump

# Avaliação do modelo
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f'MSE: {mse}')
```

```
MSE: 48171.08164548659
```

In [35]:
```
# Salvando o modelo em um arquivo .pkl
dump(model, 'modelo_previsao_preco.pkl')
```

Out[35]:
```
['modelo_previsao_preco.pkl']
```

In [36]:

In [ ]: