3 - Modelagem

In [ ]:

```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn.metrics import mean_squared_error, r2_score

import pickle
```

In [ ]:

```
df = pd.read_csv('data/df_final.csv')
```

Separando os dados em treino e teste

In [ ]:

```
encoder = LabelEncoder()
```

In [ ]:

```
def get_metrics (y_true, y_pred):
    dict_metrics = {
        'R2': metrics.r2_score(y_true, y_pred),
        'MSE': metrics.mean_squared_error(y_true, y_pred),
        'RMSE': np.sqrt(metrics.mean_squared_error(y_true, y_pred))
    }
    return dict_metrics
```

In [ ]:

```
df['bairro'] = encoder.fit_transform(df['bairro'])
df['bairro_group'] = encoder.fit_transform(df['bairro_group'])
df['room_type'] = encoder.fit_transform(df['room_type'])
```

In [ ]:

```
df.head()
```

Out[ ]:

	id	nome	host_id	host_name	bairro_group	bairro	latitude	longitude	room_type	price	mi
0	2595	Skylit Midtown Castle	2845	Jennifer	2	129	40.75362	-73.98377	0	225.0	
1	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	2	96	40.80902	-73.94190	2	150.0	
2	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	1	41	40.68514	-73.95976	0	89.0	
3	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	2	61	40.79851	-73.94399	0	80.0	
4	5099	Large Cozy 1 BR Apartment In Midtown East	7322	Chris	2	139	40.74767	-73.97500	0	200.0	

In [ ]:

```
df['latitude'] = df['latitude'].fillna(0)
```

```
df = df.loc[df['latitude'] != 0]
```

In [ ]:

```
X = df.drop(columns = ['id', 'nome', 'host_id', 'host_name', 'price', 'calculado_host_listing

y = df['price']
```

In [ ]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [ ]:

```
regressor = RandomForestRegressor()
```

In [ ]:

```
train_fit = regressor.fit(X_train,y_train)
```

In [ ]:

```
y_pred = regressor.predict(X_test)
```

In [ ]:

```
mse_rf = mean_squared_error(y_test, y_pred)
rmse_rf = np.sqrt(mse_rf)
r2_rf = r2_score(y_test, y_pred)

print(f"Random Forest - Mean Squared Error: {mse_rf}")
print(f"Random Forest - Root Mean Squared Error: {rmse_rf}")
print(f"Random Forest - R^2 Score: {r2_rf}")
```

```
Random Forest - Mean Squared Error: 169450.24759094202
Random Forest - Root Mean Squared Error: 411.6433499899422
Random Forest - R^2 Score: -0.08422987663744856
```

Infelizmente o modelo se mostrou ineficaz, apresentando métricas muito ruins para a precificação do modelo, com isso, seria necessário voltar aos df e fazer uma análise mais criteriosa dos dados.

Devido a falta de tempo, não será possível esse processo, com isso, o modelo de precificação não foi concluído.

In [ ]:

```
with open('modelo.pkl', 'wb') as f:
    pickle.dump(train_fit, f)
```