Projeto de Engenharia de Dados

Caso de Uso: Análise de Preferências de Serviços por Tipo de Hotel

Objetivo: Analisar quais serviços são mais populares por hotel e como isto afeta a preferência por parte dos clientes.

Descrição: Utilizando os dados das tabelas `hotel` e `reservas`, podemos realizar uma análise detalhada para entender as preferências de serviços entre diferentes tipos de hotéis. Isso pode ajudar os gerentes de hotel a otimizar a oferta de serviços e os desenvolvedores de plataformas de reservas a melhorar as recomendações personalizadas para os clientes.

Passos para a Análise:

1. **Carga de Dados:**
    a. Subir os dados das tabelas que estão em arquivos csv para a nuvem do Google.
    b. Usando um notebook creado no Workbench e usando linguagem python criar um pandas dataframe a partir do csv e limpar e transformar os dados (completar valores faltantes, transformar valores Yes/No a binários, etc)
    c. Subir dados limpos para uma tabela em Bigquery (Bigquery é o data warehouse de Google que será usado como banco de dados).

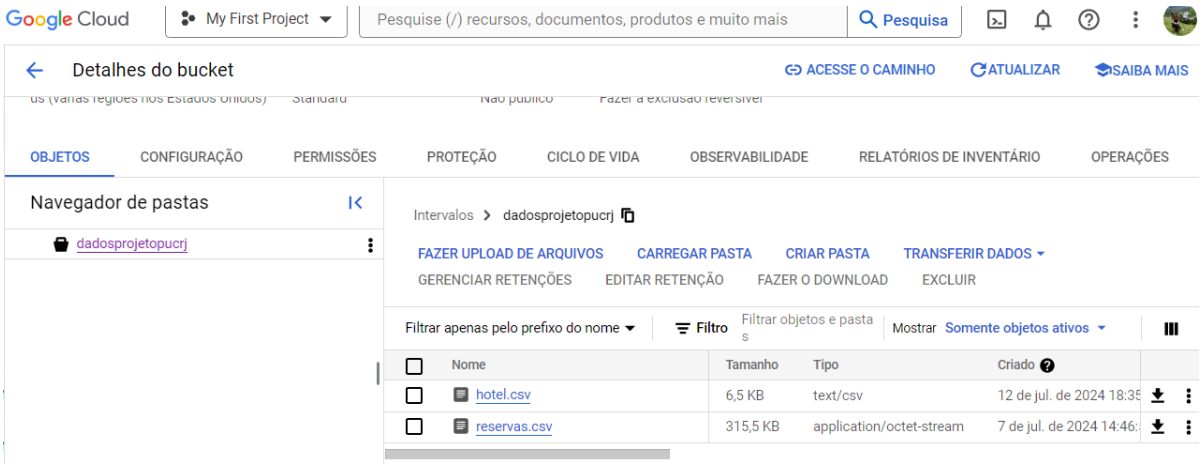


Figura 1: Interface visual da Consola de GCP do Cloud Storage, especificamente do bucket com os dados do projeto.

Os dados podem ser carregados nos buckets visualmente (como na figura) ou a nível de código, neste projeto foram subidos manualmente usando a interface visual da Consola. No Readme.md do projeto disponibiliza-se documentação sobre como criar o bucket e subir os arquivos.

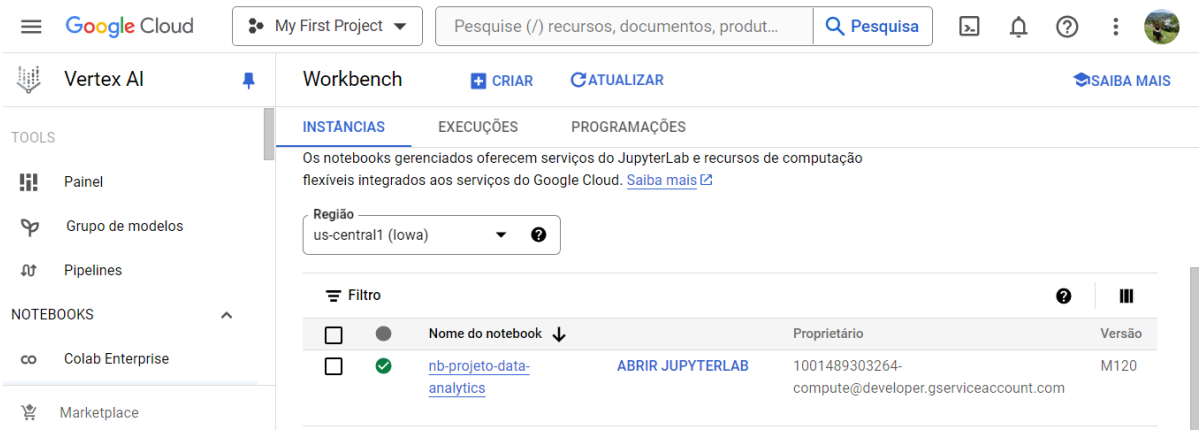


Figura 2: Interface visual do Workbench de VertexAI, onde cria-se o notebook. No Readme.md do projeto disponibiliza-se documentação sobre como criar notebooks gerenciados pelo usuário.

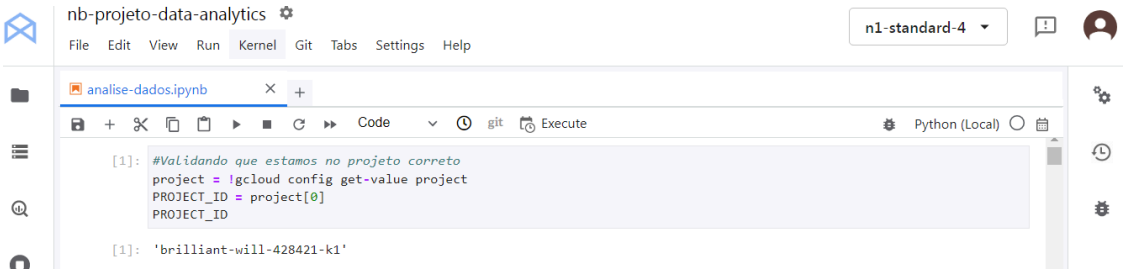


Figura 3: Interface visual do notebook do projeto.

Além da criação do bucket e a carga dos dados para o bucket, todas as etapas restantes são feitas via código desde o notebook do projeto. Para clonar o notebook e verificar que variáveis de configuração são necessárias por favor ler o Readme.md do projeto.

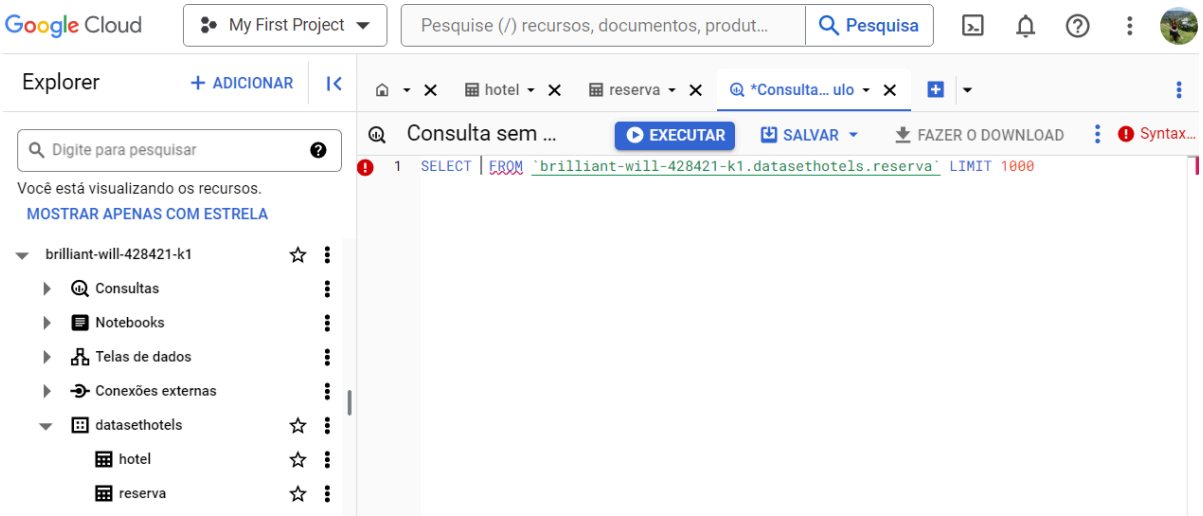


Figura 4: Interface visual do Bigquery, onde foi criado o dataset do projeto e as tabelas de dados.

As tabelas são geradas a nível de código no notebook. A Figura 5 mostra um exemplo de código de como inserir um dataframe pandas no Bigquery como uma nova tabela ou anexando os dados se a tabela existe.

```
[26]: #Carregando os dados numa tabela no Datawarehouse (Bigquery estou usando)
      job = client.load_table_from_dataframe(df_reserva, table_ref_reserva)

      # Esperar a que el trabajo se complete
      job.result()

[26]: LoadJob<project=brilliant-will-428421-k1, location=us-central1, id=9cc00e3f-8a51-4bfc-9bac-dc6ebd135ae9>
```

Figura 5: Exemplo de código de como transformar um dataframe em pandas em uma tabela em Bigquery.

2. **Descrição estatística dos dados**

	Star Rating	Rating	Free Parking	Fitness Centre	Spa and Wellness Centre	Airport Shuttle	Staff	Facilities	Location	Comfort	Cleanliness	Price Per Day($)
count	100.000000	100.000000	100.000000	100.00000	100.00000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000
mean	3.870000	8.848000	0.650000	0.21000	0.15000	0.780000	9.292000	8.821000	8.932000	9.000000	8.998000	84.050000
std	0.580056	0.758851	0.479372	0.40936	0.35887	0.416333	0.509045	0.915335	0.823258	0.731679	0.765279	64.901693
min	2.000000	5.000000	0.000000	0.00000	0.00000	0.000000	7.500000	2.500000	7.000000	5.000000	5.000000	6.000000
25%	4.000000	8.500000	0.000000	0.00000	0.00000	1.000000	9.100000	8.600000	8.375000	8.800000	8.800000	45.000000
50%	4.000000	9.000000	1.000000	0.00000	0.00000	1.000000	9.400000	9.000000	9.100000	9.100000	9.100000	75.500000
75%	4.000000	9.300000	1.000000	0.00000	0.00000	1.000000	9.625000	9.300000	9.700000	9.400000	9.400000	112.625000
max	5.000000	10.000000	1.000000	1.00000	1.00000	1.000000	10.000000	10.000000	10.000000	10.000000	10.000000	525.000000

	reserva_id	hotel_id	cliente_id	monto_total
count	5000.000000	5000.000000	5000.000000	5000.000000
mean	2499.500000	51.708400	501.072800	4975.095000
std	1443.520003	28.890523	291.146637	2886.964051
min	0.000000	1.000000	1.000000	0.000000
25%	1249.750000	27.000000	246.000000	2499.000000
50%	2499.500000	52.000000	504.000000	4949.500000
75%	3749.250000	77.000000	757.000000	7468.250000
max	4999.000000	100.000000	1000.000000	9999.000000

Figura 6: Resumo estatístico da tabela hotel (esquerda) e da tabela reserva (direita) gerados com a biblioteca pandas.

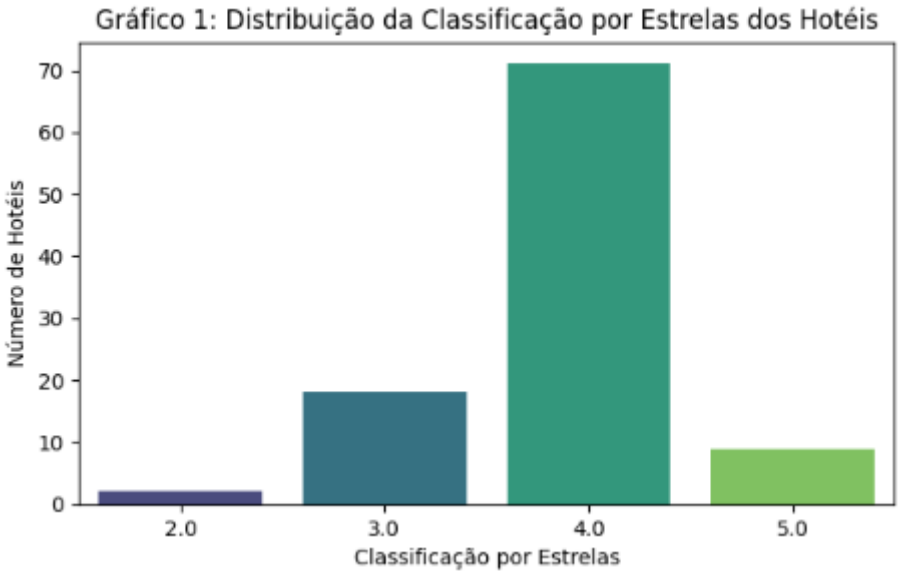
A descrição dos dados mostra que todas as variáveis em ambas tabelas não precisam ser normalizadas inicialmente porque os rangos (máximo, mínimo, média e desvio padrão) oscilam nas mesmas escalas ou em escalas próximas. Neste caso não existe o risco de variáveis com pouca informação em escalas maiores afetem a significância de outras em escalas inferiores.

3. **Visualização de Dados:**

Foram criadas visualizações como gráficos de barras e de caixa para mostrar a distribuição de preferências de serviços por tipo de hotel. Além disso, normalmente exploramos os dados para ter mais conhecimento do problema e das variáveis.

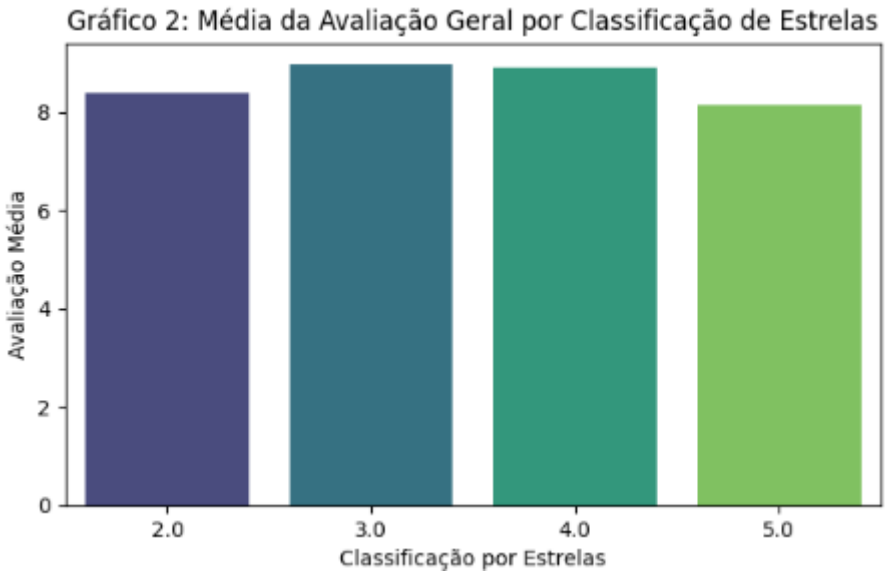
- Os hotéis mais comuns são hotéis com 4 estrelas. Na modelagem esse seria um aspecto para ser tratado já que teríamos uma classe dominante que pode afetar a modelagem e criar um viés no modelo.

```
[47]: import pandas as pd
      import matplotlib.pyplot as plt
      import seaborn as sns

      # Suponiendo que df_hotel ya está disponible y tiene las columnas renombradas

      # 1. Distribuição da classificação por estrelas dos hotéis
      plt.figure(figsize=(7, 4))
      sns.countplot(x='star_rating', data=df_hotel, palette='viridis')
      plt.title('Distribuição da Classificação por Estrelas dos Hotéis')
      plt.xlabel('Classificação por Estrelas')
      plt.ylabel('Número de Hotéis')
      plt.show()
```



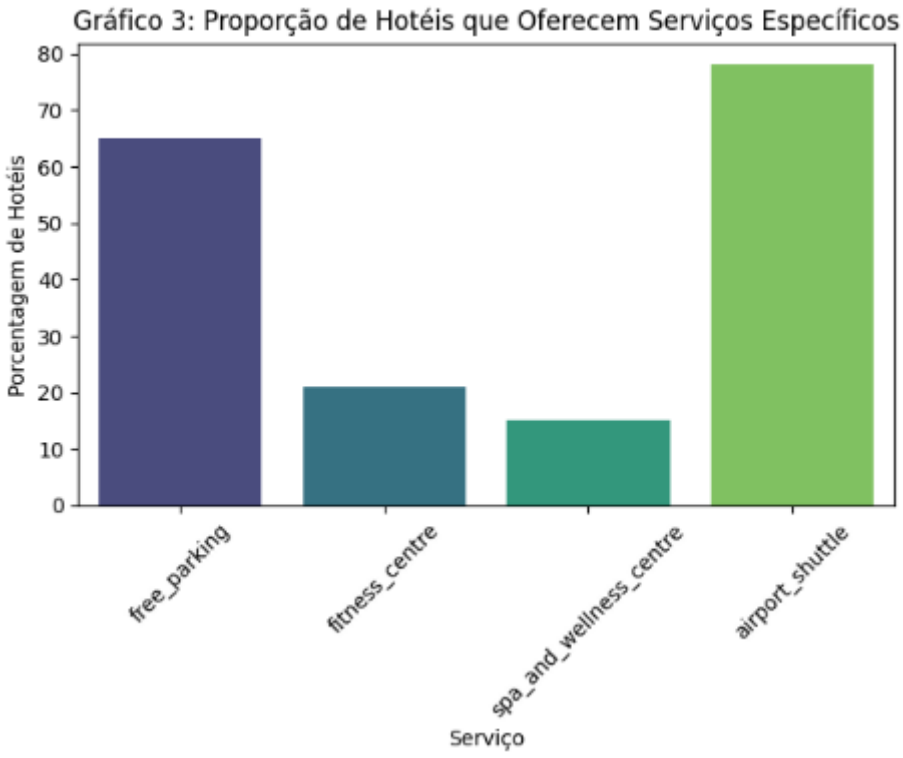
- A média de avaliação dos hotéis vs número de estrelas dá uma ideia de que o número de estrelas não é muito importante para avaliações boas dos clientes. Pode ser que neste caso sejam mais importantes outros fatores relacionados com a qualidade dos serviços.

```
# 2. Média da avaliação geral por classificação de estrelas
avg_rating_by_star = df_hotel.groupby('star_rating')['rating'].mean().reset_index()
plt.figure(figsize=(7, 4))
sns.barplot(x='star_rating', y='rating', data=avg_rating_by_star, palette='viridis')
plt.title('Média da Avaliação Geral por Classificação de Estrelas')
plt.xlabel('Classificação por Estrelas')
plt.ylabel('Avaliação Média')
plt.show()
```



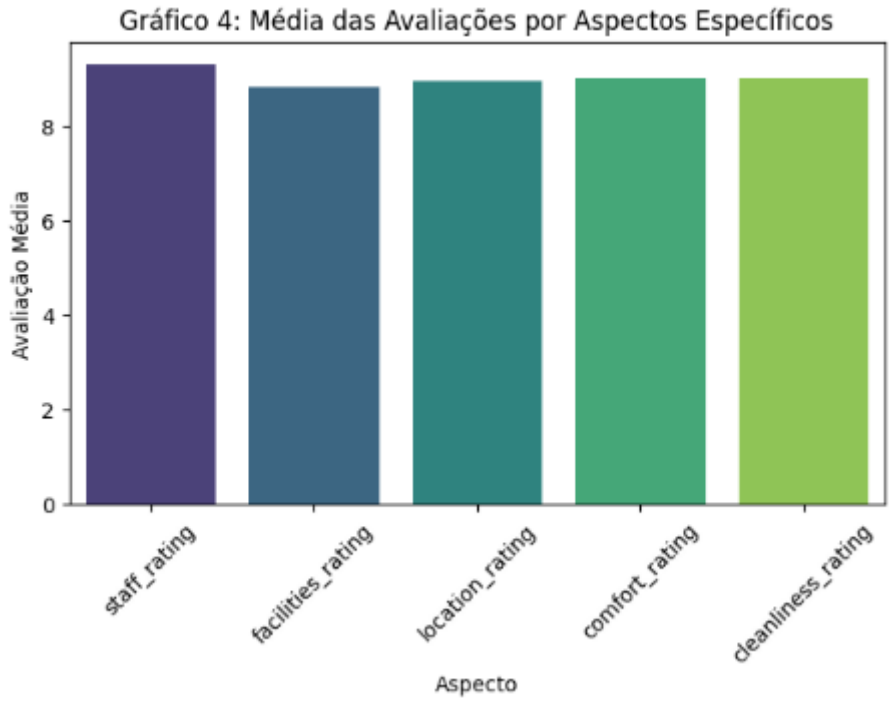
- Os serviços mais comuns são transferência para o aeroporto e estacionamento sem custos. Alguns hotéis oferecem outros serviços como spa e academia.

```
# 3. Proporção de hotéis que oferecem serviços específicos
services = ['free_parking', 'fitness_centre', 'spa_and_wellness_centre', 'airport_shuttle']
service_availability = df_hotel[services].mean().reset_index()
service_availability.columns = ['serviço', 'disponibilidade']
service_availability['disponibilidade'] *= 100  # Convertir a porcentagem

plt.figure(figsize=(7, 4))
sns.barplot(x='serviço', y='disponibilidade', data=service_availability, palette='viridis')
plt.title('Proporção de Hotéis que Oferecem Serviços Específicos')
plt.xlabel('Serviço')
plt.ylabel('Porcentagem de Hotéis')
plt.xticks(rotation=45)
plt.show()
```



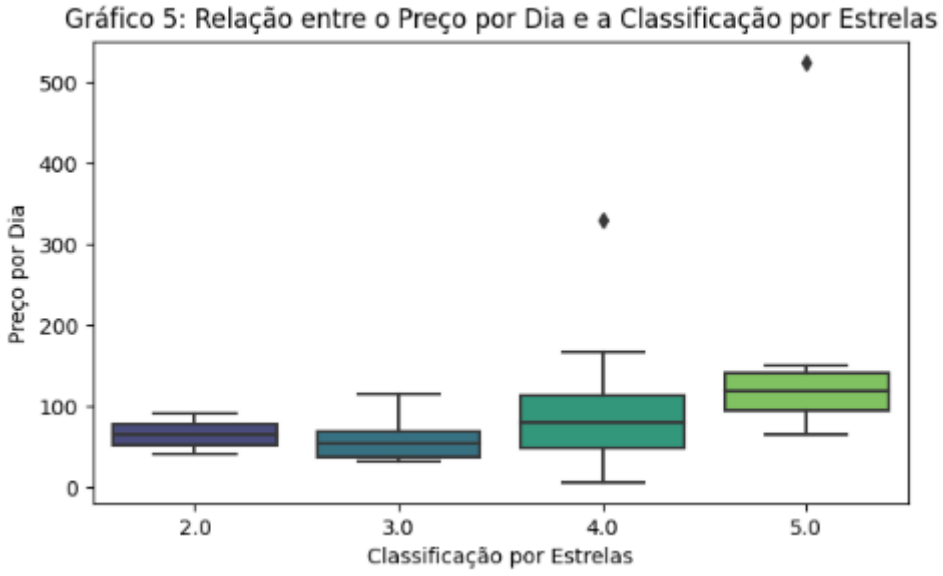

```
# 4. Média das avaliações de pessoal, instalações, localização, conforto e limpeza
ratings_columns = ['staff_rating', 'facilities_rating', 'location_rating', 'comfort_rating', 'cleanliness_rating']
avg_ratings = df_hotel[ratings_columns].mean().reset_index()
avg_ratings.columns = ['aspecto', 'avaliação_média']

plt.figure(figsize=(7, 4))
sns.barplot(x='aspecto', y='avaliação_média', data=avg_ratings, palette='viridis')
plt.title('Média das Avaliações por Aspectos Específicos')
plt.xlabel('Aspecto')
plt.ylabel('Avaliação Média')
plt.xticks(rotation=45)
plt.show()
```

- Serviços e vantagens como: capacitação dos colaboradores ,avaliações dos clientes, localização,conforto e limpeza não se mostram muito diferentes na avaliação média dos hotéis analisados.



- O gráfico de caixa mostra um resumo das distribuições de dados dos hotéis por número de estrelas, pode observar-se 2 outliers. Esses valores podem dificultar a modelagem por esse motivo é importante identificá-los antes de modelar o problema e criar um modelo que represente os dados.

```
# 5. Relação entre o preço por dia e a classificação por estrelas
plt.figure(figsize=(7, 4))
sns.boxplot(x='star_rating', y='price_per_day', data=df_hotel, palette='viridis')
plt.title('Relação entre o Preço por Dia e a Classificação por Estrelas')
plt.xlabel('Classificação por Estrelas')
plt.ylabel('Preço por Dia')
plt.show()
```



4. **Junção de dados para criar matriz de correlação entre as tabelas hotel e reserva**
    a. Unir as tabelas `hotel` e `reserva` utilizando uma chave comum, como o ID do hotel (`hotel_id`). Isso permitirá combinar informações do hotel (incluindo tipo de hotel, localização, classificação por estrelas, etc.) com as reservas realizadas.
    b. Depois foi construida uma matriz de correlação entre características dos hotéis e quantidade de reservas (`num_reservas`)

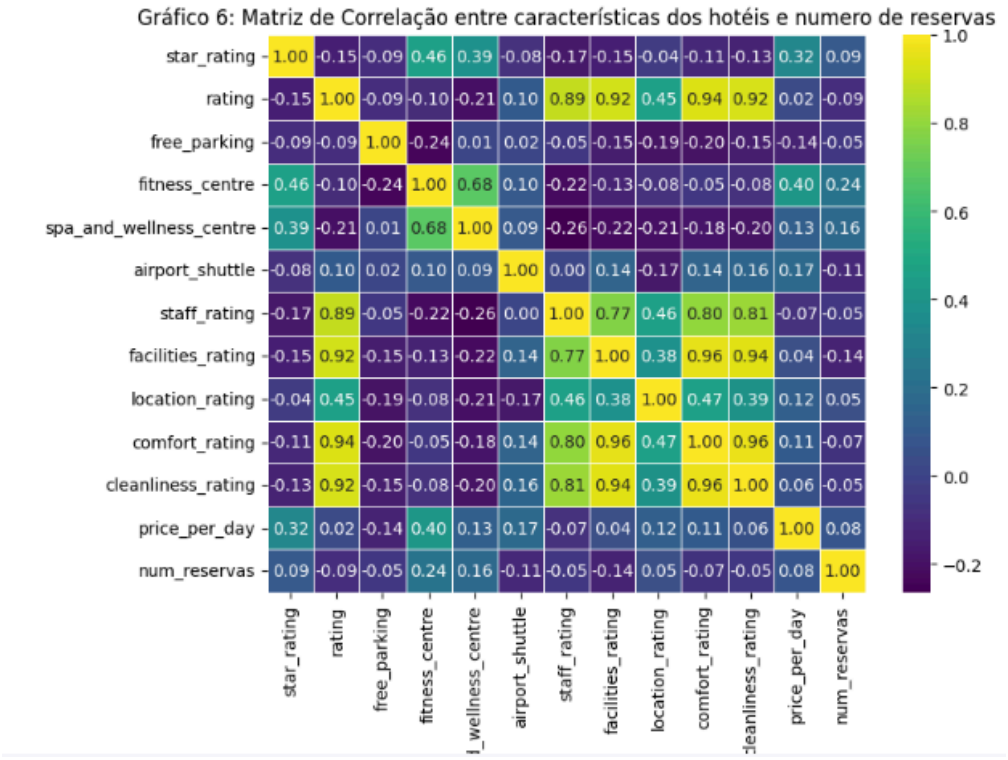
[48]:
```
from google.cloud import bigquery

# Crear una instancia de cliente de BigQuery
client = bigquery.Client()

# Definir la consulta
query = '''
    SELECT
        h.hotel_name,
        h.star_rating,
        h.rating,
        h.free_parking,
        h.fitness_centre,
        h.spa_and_wellness_centre,
        h.airport_shuttle,
        h.staff_rating,
        h.facilities_rating,
        h.location_rating,
        h.comfort_rating,
        h.cleanliness_rating,
        h.price_per_day,
        COUNT(r.reserva_id) AS num_reservas
    FROM
        `brilliant-will-428421-k1.datasethotels.hotel` h
    JOIN
        `brilliant-will-428421-k1.datasethotels.reserva` r
    ON
        h.hotel_id = r.hotel_id
    GROUP BY
        h.hotel_name,
        h.star_rating,
        h.rating,
        h.free_parking,
        h.fitness_centre,
        h.spa_and_wellness_centre,
        h.airport_shuttle,
        h.staff_rating,
        h.facilities_rating,
        h.location_rating,
        h.comfort_rating,
        h.cleanliness_rating,
        h.price_per_day
'''
```

Figura 6: Consulta que se executa para unir as duas tabelas, hotel e reserva.

```python
# Ejecutar la consulta y convertir los resultados a un DataFrame de Pandas
df = client.query(query).to_dataframe()

# 7. Matriz de correlação hotel vs cantidad de reservas
# Seleccionar solo las columnas numéricas (excluyendo la primera columna que es string)
numeric_columns = df.select_dtypes(include=['number']).columns
# Calcular la matriz de correlación entre las columnas numéricas
correlation_matrix = df[numeric_columns].corr()
# Configurar el tamaño de la figura
plt.figure(figsize=(8, 6))
# Crear el mapa de calor de la matriz de correlación
sns.heatmap(correlation_matrix, annot=True, cmap='viridis', fmt='.2f', linewidths=.5)
# Añadir título
plt.title('Matriz de Correlação entre características dos hotéis e numero de reservas')
plt.show()
```

Figura 7: Código para gerar matriz de correlação.

- Foi criada uma matriz de correlação para avaliar as relações entre variáveis.



Na matriz de correlação podem ser extraídas as relações lineares que existem entre variáveis. Por exemplo:

1. Existe uma relação forte entre a variável raiting do hotel com as variáveis staff, facilities, location, comfort e cleanliness. Isto significa que podemos ter uma ideia da avaliação que o hotel teria baseado nos valores destas variáveis. No entanto, se a variável target fosse uma variável diferente a essas listadas acima (por exemplo número de reservas), pode ser que não seja necessário que todas estejam presentes no modelo final, já que existe uma alta correlação entre elas, isso deve ser explorado.

2. Não existe uma relação forte entre as características do hotel com o número de reservas, pelo menos inicialmente não se observam relações lineares, mas pode ser que existam outros tipos de relações não lineares que podem ser exploradas.

5. Conclusões e Recomendações:

Com base na análise, é possível identificar padrões e tendências em relação às preferências de serviços por tipo de hotel.

- Propor recomendações aos gerentes de hotel sobre como ajustar a oferta de serviços para melhor atender às expectativas dos clientes e melhorar a competitividade do hotel.