```python
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt

# Carregando o arquivo que contém os dados
data = pd.read_csv("houses_to_rent_v2.csv")

# Configuração do layout do app Streamlit para que ocupe todo o espaço da tela
st.set_page_config(layout="wide")

# Cabeçalho da atividade
st.title('UNIVERSIDADE FEDERAL DO MARANHÃO - UFMA')

st.text("Especialização em Análise de Dados e Inteligência Artificial")
st.text("Disciplina: Visualização de Dados")
st.text("Aluno: Pedro Wemerson Pereira da Silva")
st.text("Email: pedrowemersonp@gmail.com")

# Título do Dashboard
st.title("Dashboard de Imóveis para Aluguel")

# Criando uma caixa de seleção para que o usuário possa ver todas para ver as informações se preferir
@st.cache_data
def load_data(nrows):
    data = pd.read_csv("houses_to_rent_v2.csv", nrows=nrows)
    lowercase = lambda x: str(x).lower()
    data.rename(lowercase, axis='columns', inplace=True)
    return data

if st.checkbox('Clique aqui para exibir a base de dados'):
    st.subheader('Base de Dados')
    st.write(data)

# Mensagem para melhorar a compreensão por parte do usuário
st.text('Interaja com o Dashboard abaixo para filtrar as informações')

# Criando caixa de seleção para selecionar a quantidade de quartos
rooms_selected = st.selectbox("Escolha a quantidade de quartos", sorted(data['rooms'].unique()))

# Aplicando filtro para a quantidade de quartos selecionada
filtered_data = data[data['rooms'] == rooms_selected]

# Criar um layout de 3 colunas para exibir a quantidade de imóveis, o menor valor e maior valor
col1, col2, col3 = st.columns(3)
```

```
# Exibir os dados nas respectivas colunas
with col1:
    total_properties = len(filtered_data)
    st.metric(label="Quantidade de imóveis", value=total_properties)

with col2:
    min_rent = filtered_data['rent amount (R$)'].min()
    st.metric(label="Menor Valor", value=f"R$ {min_rent}")

with col3:
    max_rent = filtered_data['rent amount (R$)'].max()
    st.metric(label="Maior Valor", value=f"R$ {max_rent}")

# Média dos preços por cidade (em ordem decrescente)
avg_price_by_city = filtered_data.groupby('city')['rent amount
(R$)'].mean().sort_values(ascending=True)

# Média dos preços dos imóveis mobiliados e não mobiliados
avg_price_furnished = filtered_data[filtered_data['furniture'] ==
'furnished']['rent amount (R$)'].mean()
avg_price_not_furnished = filtered_data[filtered_data['furniture'] ==
'not furnished']['rent amount (R$)'].mean()

# Gráfico de barras para a média de preços por cidade em ordem
decrescente
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(18, 8),
gridspec_kw={'width_ratios': [2, 1]})

# Gráfico de barras horizontal para a média dos preços por cidade
ax1.barh(avg_price_by_city.index, avg_price_by_city.values,
color='black', edgecolor='none')
ax1.set_title("Preço Médio por Cidade (Ordem Decrescente)")
ax1.spines['top'].set_visible(False)
ax1.spines['right'].set_visible(False)
ax1.spines['left'].set_visible(False)
ax1.spines['bottom'].set_visible(False)
ax1.tick_params(left=False, bottom=False)
ax1.set_xticks([])
for i, v in enumerate(avg_price_by_city.values):
    ax1.text(v + 100, i, f'{v:.2f}', va='center', color='black')

# Gráfico de barras para a média de preços dos imóveis mobiliados e não
mobiliados
ax2.bar(['Com Mobília', 'Sem Mobília'], [avg_price_furnished,
avg_price_not_furnished], color=['grey', 'grey'], edgecolor='none')
ax2.set_title("Preço Médio com e sem Mobília")
ax2.spines['top'].set_visible(False)
```

```
ax2.spines['right'].set_visible(False)
ax2.spines['left'].set_visible(False)
ax2.spines['bottom'].set_visible(False)
ax2.tick_params(left=False, bottom=False)  # Remover ticks do eixo y
ax2.set_yticks([])  # Remover valores do eixo y
for i, v in enumerate([avg_price_furnished, avg_price_not_furnished]):
    ax2.text(i, v + 100, f'{v:.2f}', ha='center', color='black')

# Exibindo os gráficos usando a largura total da página
st.pyplot(fig)

# Pegar a cidade com o maior e menor valor
cidade_maior_preco = avg_price_by_city.index[0]
cidade_menor_preco = avg_price_by_city.index[-1]

# Adicionar um caixa de seleção para seleção da cidade, já selecionando
a cidade com o maior valor
city_selected = st.selectbox("Escolha a cidade",
sorted(filtered_data['city'].unique()), index=0)

# Filtrar os imóveis pela cidade selecionada e ordenar por preço
decrescente
city_filtered_data = filtered_data[filtered_data['city'] ==
city_selected].sort_values(by='rent amount (R$)', ascending=False)

# Exibir a lista dos primeiros imóveis da cidade selecionada, sem o
índice
st.write(f"Imóveis na cidade de {city_selected} (ordenados por preço de
forma decrescente):")
st.dataframe(city_filtered_data[['city', 'rent amount (R$)', 'rooms',
'furniture', 'total (R$)']].head(10), width=1400,
height=300,hide_index=True)
```