

- ✓ Definindo algumas funções

- ✓ 1. Função para calcular o faturamento das lojas

```
# Função para calcular o faturamento das lojas e adiciona-las ao DataFrame dados
def calcular_faturamento(loja, x):
    faturamento = loja['Preço'].sum()
    dados_faturamento.loc[x-1] = [f'Loja {x}', faturamento]
    print(f'Faturamento Loja {x}: R$ {faturamento:.2f}')
    return faturamento
```

- ✓ 2. Função para ler e gerar um dataframe acerca das vendas por categoria

```
def vendas_categoria(loja):
    qtd_categoria = loja['Categoria do Produto'].value_counts().reset_index()
    qtd_categoria.columns = ['Categoria', 'Quantidade']
    qtd_categoria['Porcentagem'] = round(qtd_categoria['Quantidade'] / qtd_categoria['Quantidade'].sum() * 100, 2)

    return qtd_categoria
```

- ✓ 3. Função para calcular a média de avaliação das lojas

```
def media_avaliacao(loja, x):
    avaliacao = loja['Avaliação da compra'].mean()
    avaliacao_media.loc[x-1] = [f'Loja {x}', round(avaliacao, 2)]

    return avaliacao
```

- ✓ 4. Função para calcular o frete médio, os fretes solicitados e os não solicitados das lojas

```
def media_frete(loja, x):
    frete = round(loja['Frete'].mean(), 2)
    frete_medio.loc[x-1, ['Loja', 'Frete Médio']] = [f'Loja {x}', frete]

    return (frete, frete_medio)
```

- ✓ Função para calcular os produtos mais vendidos e os menos vendidos

```
def prod_mais_vendidos(loja):
    mais_vendidos = loja['Produto'].value_counts().reset_index()
    mais_vendidos.columns = ['Produto', 'Quantidade']

    return mais_vendidos.head(5)
```

```
def prod_menos_vendidos(loja):
    menos_vendidos = loja['Produto'].value_counts().sort_values(ascending= True).reset_index()
    menos_vendidos.columns = ['Produto', 'Quantidade']

    return menos_vendidos.head(5)
```

- ✓ Importação dos dados

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
url_1 = "https://raw.githubusercontent.com/alura-es-cursos/challenge1-data-science/refs/heads/main/base-de-dados-challenge"
url_2 = "https://raw.githubusercontent.com/alura-es-cursos/challenge1-data-science/refs/heads/main/base-de-dados-challenge"
url_3 = "https://raw.githubusercontent.com/alura-es-cursos/challenge1-data-science/refs/heads/main/base-de-dados-challenge"
url_4 = "https://raw.githubusercontent.com/alura-es-cursos/challenge1-data-science/refs/heads/main/base-de-dados-challenge"
```

- ✓ Realizando a leitura dos dados das lojas

```
loja_1 = pd.read_csv(url_1)
loja_2 = pd.read_csv(url_2)
loja_3 = pd.read_csv(url_3)
loja_4 = pd.read_csv(url_4)
```

- ✓ 1. Análise do faturamento
- ✓ Tratando alguns dados
- ✓ Colocando as primeiras letras das categorias dos produtos em maiúsculas

```
loja_1['Categoria do Produto'] = [s.capitalize() for s in loja_1['Categoria do Produto']]
loja_2['Categoria do Produto'] = [s.capitalize() for s in loja_2['Categoria do Produto']]
loja_3['Categoria do Produto'] = [s.capitalize() for s in loja_3['Categoria do Produto']]
loja_4['Categoria do Produto'] = [s.capitalize() for s in loja_4['Categoria do Produto']]
```

- ✓ Transformando a coluna 'Data da Compra' que está em formato 'object' para 'datetime'

```
loja_1['Data da Compra'] = pd.to_datetime(loja_1['Data da Compra'], dayfirst=True)
loja_2['Data da Compra'] = pd.to_datetime(loja_2['Data da Compra'], dayfirst=True)
loja_3['Data da Compra'] = pd.to_datetime(loja_3['Data da Compra'], dayfirst=True)
loja_4['Data da Compra'] = pd.to_datetime(loja_4['Data da Compra'], dayfirst=True)
```

- ✓ Criando o DataFrame para análise das lojas Alura Store

```
# Criação do DataFrame para análise do faturamento das lojas Alura Store
dados_faturamento = pd.DataFrame(columns=['Loja', 'Faturamento'])
```

```
calcular_faturamento(loja_1, 1)
calcular_faturamento(loja_2, 2)
calcular_faturamento(loja_3, 3)
calcular_faturamento(loja_4, 4)
```

[Mostrar saída oculta](#)

```
dados_faturamento['Porcentagem'] = round(dados_faturamento['Faturamento'] / dados_faturamento['Faturamento'].sum() * 100)
dados_faturamento
```

[Mostrar saída oculta](#)

- ✓ Gerando uma visualização do faturamento nas 4 unidades da Alura Store em um gráfico de barras

```
fig = plt.figure(figsize=(10, 6), facecolor="#cccccc")
ax = plt.axes()
ax.set_facecolor('#cccccc')
bars = plt.bar(dados_faturamento['Loja'], dados_faturamento['Faturamento'], color='#203f75')

for bar in bars:
    x_pos = bar.get_x() + bar.get_width() / 2
    y_pos = bar.get_height() + (dados_faturamento['Faturamento'].max() * 0.01)
    value_text = f'R$ {bar.get_height()/1e0:.2f}'
    plt.text(x_pos, y_pos, value_text, ha='center', va='bottom', color='#19325b', fontsize=12)

formatted_yticks = ['R$ {:.2f} M'.format(val/1e6) for val in ax.get_yticks()]
ax.set_yticks(ax.get_yticks())
ax.set_yticklabels(formatted_yticks, fontsize=12, color='#19325b')

plt.title('Faturamento das Unidades Alura Store', color='#19325b', fontsize= 18)
plt.xlabel('')
plt.ylabel('')
plt.xticks(fontsize= 12, color='#19325b')
plt.yticks(fontsize= 12, color='#19325b')

ax.spines[['top', 'right']].set_visible(False)
ax.yaxis.grid(color='gray', linestyle='--', linewidth= 0.5)
ax.set_axisbelow(True)

plt.show()
```

[Mostrar saída oculta](#)

✓ 2. Vendas por categoria

✓ Gerando os DataFrames e gráficos de forma individual

```
cat_loja_1 = vendas_categoria(loja_1)
cat_loja_1
```

[Mostrar saída oculta](#)

```
fig, ax = plt.subplots(figsize=(10, 4))
ax = sns.barplot(data=cat_loja_1, y=cat_loja_1['Categoria'], x='Quantidade', orient='h', hue='Quantidade', palette='coolwarm')

for i, v in enumerate(cat_loja_1['Quantidade']):
    ax.text(v + 1, i, str(v), color='#19325b', fontsize=10, ha = 'left', va = 'center')

ax.set_title(f'Maiores Vendas por Categoria\nAlura Store - Loja 1', loc='center', fontsize=18)
ax.set_xlabel('')
ax.set_ylabel('')

ax.set_frame_on(False)
ax.get_xaxis().set_visible(False)
ax.tick_params(axis='both', which='both', length= 0)
ax.spines[['top', 'right', 'bottom']].set_visible(False)

plt.show()
```

[Mostrar saída oculta](#)

```
cat_loja_2 = vendas_categoria(loja_2)
```

[Mostrar saída oculta](#)

```
fig, ax = plt.subplots(figsize=(10, 4))
ax = sns.barplot(data=cat_loja_2, y=cat_loja_2['Categoria'], x='Quantidade', orient='h', hue='Quantidade', palette='coolwarm')

for i, v in enumerate(cat_loja_2['Quantidade']):
    ax.text(v + 1, i, str(v), color='#19325b', fontsize=10, ha = 'left', va = 'center')

ax.set_title(f'Maiores Vendas por Categoria\nAlura Store - Loja 2', loc='center', fontsize=18)
ax.set_xlabel('')
ax.set_ylabel('')

ax.set_frame_on(False)
ax.get_xaxis().set_visible(False)
ax.tick_params(axis='both', which='both', length= 0)
ax.spines[['top', 'right', 'bottom']].set_visible(False)

plt.show()
```

[Mostrar saída oculta](#)

```
cat_loja_3 = vendas_categoria(loja_3)
```

[Mostrar saída oculta](#)

```
fig, ax = plt.subplots(figsize=(10, 4))
ax = sns.barplot(data=cat_loja_3, y=cat_loja_3['Categoria'], x='Quantidade', orient='h', hue='Quantidade', palette='coolwarm')

for i, v in enumerate(cat_loja_3['Quantidade']):
    ax.text(v + 1, i, str(v), color='#19325b', fontsize=10, ha = 'left', va = 'center')

ax.set_title(f'Maiores Vendas por Categoria\nAlura Store - Loja 3', loc='center', fontsize=18)
ax.set_xlabel('')
ax.set_ylabel('')

ax.set_frame_on(False)
ax.get_xaxis().set_visible(False)
ax.tick_params(axis='both', which='both', length= 0)
ax.spines[['top', 'right', 'bottom']].set_visible(False)

plt.show()
```

[Mostrar saída oculta](#)

```
cat_loja_4 = vendas_categoria(loja_4)
```

[Mostrar saída oculta](#)

```

fig, ax = plt.subplots(figsize=(10, 4))
ax = sns.barplot(data=cat_loja_4, y=cat_loja_4['Categoria'], x='Quantidade', orient='h', hue='Quantidade', palette='coolwarm')
for i, v in enumerate(cat_loja_4['Quantidade']):
    ax.text(v + 1, i, str(v), color='#19325b', fontsize=10, ha = 'left', va = 'center')

ax.set_title(f'Maiores Vendas por Categoria\nAlura Store - Loja 4', loc='center', fontsize=18)
ax.set_xlabel('')
ax.set_ylabel('')

ax.set_frame_on(False)
ax.get_xaxis().set_visible(False)
ax.tick_params(axis='both', which='both', length= 0)
ax.spines[['top', 'right', 'bottom']].set_visible(False)

plt.show()

```

[Mostrar saída oculta](#)

▼ Concatenando os dados das 4 lojas e gerando um gráfico

```

loja_1['Loja'] = 'Loja 1'
loja_2['Loja'] = 'Loja 2'
loja_3['Loja'] = 'Loja 3'
loja_4['Loja'] = 'Loja 4'

```

```
df_total = pd.concat([loja_1, loja_2, loja_3, loja_4], ignore_index=True)
```

```
vendas_categoria_final = pd.crosstab(index=df_total['Loja'], columns=df_total['Categoria do Produto'])
```

```
vendas_categoria_final
```

[Mostrar saída oculta](#)

```

# Criação de um gráfico de barras agrupadas para exibir as vendas por categoria e loja
df_plot = vendas_categoria_final.reset_index()
df_plt_vendas = df_plot.melt(id_vars='Loja', var_name='Categoria do Produto', value_name='Quantidade')

plt.figure(figsize=(14, 6))
sns.barplot(data=df_plt_vendas, x='Categoria do Produto', y='Quantidade', hue='Loja')
plt.title('Quantidade de Produtos Vendidos por Categoria e Loja', fontsize= 18)
plt.xlabel('Categoria do Produto', fontsize= 14)
plt.ylabel('Quantidade Vendida', fontsize= 14)
plt.xticks(rotation=0, ha='center')
plt.legend(title='Loja')
plt.tight_layout()
plt.show()

```

[Mostrar saída oculta](#)

▼ 3. Média de avaliação por loja

```
avaliacao_media = pd.DataFrame(columns=['Loja', 'Média'])
```

[Mostrar saída oculta](#)

```

media_avaliacao(loja_1, 1)
media_avaliacao(loja_2, 2)
media_avaliacao(loja_3, 3)
media_avaliacao(loja_4, 4)

```

```
np.float64(3.9957591178965224)
```

```
avaliacao_media
```

[Mostrar saída oculta](#)

```

fig = plt.figure(figsize=(6, 4), facecolor='#cccccc')
ax = plt.axes()
ax.set_facecolor('#cccccc')
bars = plt.bar(avaliacao_media['Loja'], avaliacao_media['Média'], color='#203f75')

for bar in bars:
    x_pos = bar.get_x() + bar.get_width() / 2

```

```

y_pos = bar.get_height() + (avaliacao_media['Média'].max() * 0.01)
value_text = f'{bar.get_height():.2f}'
plt.text(x_pos, y_pos, value_text, ha='center', va='bottom', color='#19325b', fontsize=10)

plt.title('Média das Avaliações das Unidades Alura Store', color='#19325b', fontsize= 12)
plt.xlabel('')
plt.xticks(fontsize= 12, color='#19325b')

ax.spines[['top', 'right']].set_visible(False)
ax.yaxis.grid(color='gray', linestyle='--', linewidth= 0.5)
ax.set_axisbelow(True)

plt.show()

```

[Mostrar saída oculta](#)

▼ 4. Produtos

```
df_produtos = pd.crosstab(index=df_total['Loja'], columns=df_total['Produto'])
```

[Mostrar saída oculta](#)

▼ 4.1 - Mais vendidos

```

# Criando uma lista para armazenar os dados organizados
dados_ranking = []

for index, row in df_produtos.iterrows():
    # Pega os 5 maiores para a loja atual
    top5 = row.nlargest(5)

    # Cria um dicionário com os nomes e valores
    linha = {
        'Loja': index,
        '1º Produto': top5.index[0], '1º Qtd': top5.values[0],
        '2º Produto': top5.index[1], '2º Qtd': top5.values[1],
        '3º Produto': top5.index[2], '3º Qtd': top5.values[2],
        '4º Produto': top5.index[3], '4º Qtd': top5.values[3],
        '5º Produto': top5.index[4], '5º Qtd': top5.values[4]
    }
    dados_ranking.append(linha)

# Transforma em DataFrame
df_final = pd.DataFrame(dados_ranking).set_index('Loja')

df_final

```

[Mostrar saída oculta](#)

```
mais_vendidos_loja_1 = prod_mais_vendidos(loja_1)
```

[Mostrar saída oculta](#)

```

fig, ax = plt.subplots(figsize=(10, 4), facecolor='#cccccc')
ax.set_facecolor('#cccccc')
ax = sns.barplot(data=mais_vendidos_loja_1, y=mais_vendidos_loja_1['Produto'], x='Quantidade', orient= 'h', hue='Quantidade')

for i, v in enumerate(mais_vendidos_loja_1['Quantidade']):
    ax.text(v + 1, i, str(v), color='#203f75', fontsize=10, ha= 'left', va='center')

ax.set_title('Produtos Mais Vendidos Alura Store\nLoja 1', loc='center', fontsize=18, color='#203f75')
ax.set_xlabel('')
ax.set_ylabel('')

ax.set_frame_on(False)
ax.get_xaxis().set_visible(False)
ax.tick_params(axis='y', which='both', length= 0, colors='#203f75')
ax.spines[['top', 'right', 'bottom']].set_visible(False)

plt.show()

```

[Mostrar saída oculta](#)

```
mais_vendidos_loja_2 = prod_mais_vendidos(loja_2)
```

```

fig, ax = plt.subplots(figsize=(10, 4), facecolor="#cccccc")
ax.set_facecolor('#cccccc')
ax = sns.barplot(data=mais_vendidos_loja_2, y=mais_vendidos_loja_2['Produto'], x='Quantidade', orient= 'h', hue='Quantidade')

for i, v in enumerate(mais_vendidos_loja_2['Quantidade']):
    ax.text(v + 1, i, str(v), color="#203f75", fontsize=10, ha= 'left', va='center')

ax.set_title('Produtos Mais Vendidos Alura Store\nLoja 2', loc='center', fontsize=18, color='#203f75')
ax.set_xlabel('')
ax.set_ylabel('')

ax.set_frame_on(False)
ax.get_xaxis().set_visible(False)
ax.tick_params(axis='y', which='both', length= 0, colors='#203f75')
ax.spines[['top', 'right', 'bottom']].set_visible(False)

plt.show()

```

[Mostrar saída oculta](#)

```
mais_vendidos_loja_3 = prod_mais_vendidos(loja_3)
```

```

fig, ax = plt.subplots(figsize=(10, 4), facecolor="#cccccc")
ax.set_facecolor('#cccccc')
ax = sns.barplot(data=mais_vendidos_loja_3, y=mais_vendidos_loja_3['Produto'], x='Quantidade', orient= 'h', hue='Quantidade')

for i, v in enumerate(mais_vendidos_loja_3['Quantidade']):
    ax.text(v + 1, i, str(v), color="#203f75", fontsize=10, ha= 'left', va='center')

ax.set_title('Produtos Mais Vendidos Alura Store\nLoja 3', loc='center', fontsize=18, color='#203f75')
ax.set_xlabel('')
ax.set_ylabel('')

ax.set_frame_on(False)
ax.get_xaxis().set_visible(False)
ax.tick_params(axis='y', which='both', length= 0, colors='#203f75')
ax.spines[['top', 'right', 'bottom']].set_visible(False)

plt.show()

```

[Mostrar saída oculta](#)

```
mais_vendidos_loja_4 = prod_mais_vendidos(loja_4)
```

```

fig, ax = plt.subplots(figsize=(10, 4), facecolor="#cccccc")
ax.set_facecolor('#cccccc')
ax = sns.barplot(data=mais_vendidos_loja_4, y=mais_vendidos_loja_4['Produto'], x='Quantidade', orient= 'h', hue='Quantidade')

for i, v in enumerate(mais_vendidos_loja_4['Quantidade']):
    ax.text(v + 1, i, str(v), color="#203f75", fontsize=10, ha= 'left', va='center')

ax.set_title('Produtos Mais Vendidos Alura Store\nLoja 4', loc='center', fontsize=18, color='#203f75')
ax.set_xlabel('')
ax.set_ylabel('')

ax.set_frame_on(False)
ax.get_xaxis().set_visible(False)
ax.tick_params(axis='y', which='both', length= 0, colors='#203f75')
ax.spines[['top', 'right', 'bottom']].set_visible(False)

plt.show()

```

[Mostrar saída oculta](#)

4.2 - Menos vendidos

```

# Criando o DataFrame formatado para os MENOS vendidos
df_menos_vendidos = df_produtos.apply(
    lambda s: [f'{prod}: {val}' for prod, val in s.nsmallest(3).items()],
    axis=1,
    result_type='expand'
)

# Renomeando as colunas para refletir o "ranking inverso"

```

```
df_menos_vendidos.columns = ['Último Lugar', 'Penúltimo Lugar', 'Antepenúltimo Lugar']  
df_menos_vendidos
```

[Mostrar saída oculta](#)

```
menos_vendidos_loja_1 = prod_menos_vendidos(loja_1)
```

```
fig, ax = plt.subplots(figsize=(10, 4), facecolor='#cccccc')  
ax.set_facecolor('#cccccc')  
ax = sns.barplot(data=menos_vendidos_loja_1, y=menos_vendidos_loja_1['Produto'], x='Quantidade', orient= 'h', hue='Quantidade')  
  
for i, v in enumerate(menos_vendidos_loja_1['Quantidade']):  
    ax.text(v + 1, i, str(v), color="#203f75", fontsize=10, ha= 'left', va='center')  
  
ax.set_title('Produtos menos Vendidos Alura Store\nLoja 1', loc='center', fontsize=18, color="#203f75")  
ax.set_xlabel('')  
ax.set_ylabel('')  
  
ax.set_frame_on(False)  
ax.get_xaxis().set_visible(False)  
ax.tick_params(axis='y', which='both', length= 0, colors='#203f75')  
ax.spines[['top', 'right', 'bottom']].set_visible(False)  
  
plt.show()
```

[Mostrar saída oculta](#)

```
menos_vendidos_loja_2 = prod_menos_vendidos(loja_2)
```

```
fig, ax = plt.subplots(figsize=(10, 4), facecolor='#cccccc')  
ax.set_facecolor('#cccccc')  
ax = sns.barplot(data=menos_vendidos_loja_2, y=menos_vendidos_loja_2['Produto'], x='Quantidade', orient= 'h', hue='Quantidade')  
  
for i, v in enumerate(menos_vendidos_loja_2['Quantidade']):  
    ax.text(v + 1, i, str(v), color="#203f75", fontsize=10, ha= 'left', va='center')  
  
ax.set_title('Produtos menos Vendidos Alura Store\nLoja 2', loc='center', fontsize=18, color="#203f75")  
ax.set_xlabel('')  
ax.set_ylabel('')  
  
ax.set_frame_on(False)  
ax.get_xaxis().set_visible(False)  
ax.tick_params(axis='y', which='both', length= 0, colors='#203f75')  
ax.spines[['top', 'right', 'bottom']].set_visible(False)  
  
plt.show()
```

[Mostrar saída oculta](#)

```
menos_vendidos_loja_3 = prod_menos_vendidos(loja_3)
```

```
fig, ax = plt.subplots(figsize=(10, 4), facecolor='#cccccc')  
ax.set_facecolor('#cccccc')  
ax = sns.barplot(data=menos_vendidos_loja_3, y=menos_vendidos_loja_3['Produto'], x='Quantidade', orient= 'h', hue='Quantidade')  
  
for i, v in enumerate(menos_vendidos_loja_3['Quantidade']):  
    ax.text(v + 1, i, str(v), color="#203f75", fontsize=10, ha= 'left', va='center')  
  
ax.set_title('Produtos menos Vendidos Alura Store\nLoja 3', loc='center', fontsize=18, color="#203f75")  
ax.set_xlabel('')  
ax.set_ylabel('')  
  
ax.set_frame_on(False)  
ax.get_xaxis().set_visible(False)  
ax.tick_params(axis='y', which='both', length= 0, colors='#203f75')  
ax.spines[['top', 'right', 'bottom']].set_visible(False)  
  
plt.show()
```

[Mostrar saída oculta](#)

```
menos_vendidos_loja_4 = prod_menos_vendidos(loja_4)
```

```
fig, ax = plt.subplots(figsize=(10, 4), facecolor='#cccccc')
ax.set_facecolor('#cccccc')
ax = sns.barplot(data=menos_vendidos_loja_4, y=menos_vendidos_loja_4['Produto'], x='Quantidade', orient= 'h', hue='Quantidade')

for i, v in enumerate(menos_vendidos_loja_4['Quantidade']):
    ax.text(v + 1, i, str(v), color='#203f75', fontsize=10, ha= 'left', va='center')

ax.set_title('Produtos menos Vendidos Alura Store\nLoja 4', loc='center', fontsize=18, color='#203f75')
ax.set_xlabel('')
ax.set_ylabel('')

ax.set_frame_on(False)
ax.get_xaxis().set_visible(False)
ax.tick_params(axis='y', which='both', length= 0, colors='#203f75')
ax.spines[['top', 'right', 'bottom']].set_visible(False)

plt.show()
```

[Mostrar saída oculta](#)

▼ 5. Frete médio por loja

```
frete_medio = pd.DataFrame(columns=['Loja', 'Frete Médio'])
```

```
media_frete(loja_1, 1)
media_frete(loja_2, 2)
media_frete(loja_3, 3)
media_frete(loja_4, 4)
```

[Mostrar saída oculta](#)

```
frete_medio
```

[Mostrar saída oculta](#)

```
fig = plt.figure(figsize=(6, 4), facecolor='#cccccc')
ax = plt.axes()
ax.set_facecolor('#cccccc')
bars = plt.bar(frete_medio['Loja'], frete_medio['Frete Médio'], color='#203f75')

for bar in bars:
    x_pos = bar.get_x() + bar.get_width() / 2
    y_pos = bar.get_height() + (frete_medio['Frete Médio'].max() * 0.01)
    value_text = f'{bar.get_height():,.2f}'
    plt.text(x_pos, y_pos, value_text, ha='center', va='bottom', color='#19325b', fontsize=10)

plt.title('Fretes das Lojas Alura Store', color='#19325b', fontsize= 12)
plt.xlabel('')
plt.xticks(fontsize= 12, color='#19325b')

ax.set_frame_on(False)
ax.get_yaxis().set_visible(False)

plt.show()
```

[Mostrar saída oculta](#)