

# analise\_ofertas\_relampago\_v2

February 13, 2023

## 1 Análise Exploratória - Ofertas relâmpago

### 1.1 1) Introdução

As ofertas relâmpago são um tipo de oferta que possui uma duração pré-definida de algumas horas e também um percentual de unidades (estoque) preparados para essas vendas

### 1.2 2) Objetivo da análise

O objetivo dessa análise exploratória é encontrar insights sobre esse tipo de oferta, como: - Comparar feriados e dias normais - Comparar dias da semana - Comparar períodos do dia - Ter uma visão geral da quantidade de vendas e valor vendido - Ter uma visão geral por categoria

## 2 3) Conjunto de dados

Esse conjunto de dados foi fornecido para a realização do teste e é composto pelas seguintes informações:

- OFFER\_START\_DATE : Data de início da venda relâmpago
- OFFER\_START\_DTTM : Data e hora de início da venda relâmpago
- OFFER\_FINISH\_DTTM : Data e hora de fim da venda relâmpago
- OFFER\_TYPE : Tipo de venda
- INVOLVED\_STOCK : Qtd. de produtos separados para a venda relâmpago
- REMAINING\_STOCK\_AFTER\_END : Qtd de produtos restantes após a venda relâmpago
- SOLD\_AMOUNT : Total em R\$ em vendas por produto
- SOLD\_QUANTITY : Quantidade vendida por produto
- ORIGIN :
- SHIPPING\_PAYMENT\_TYPE : Forma de pagamento do frete
- DOM\_DOMAIN\_AGG1 : Grupo do produto
- VERTICAL : Categoria do produto
- DOMAIN\_ID : Identificador do produto

### 2.0.1 3.2 - Preparação do dataset

```
[1]: import pandas as pd
from datetime import datetime
import extract_holidays as ext
import os
import numpy as np
```

```
import plotly.express as px
import plotly.graph_objects as go
import locale
```

```
[2]: locale.setlocale(locale.LC_ALL, "pt_BR.utf8")
```

```
[2]: 'pt_BR.utf8'
```

```
[3]: df_vendas_relampago = pd.read_csv("datasets/ofertas_relampago.csv",
    ↳ parse_dates=['OFFER_START_DATE', 'OFFER_START_DTTM', 'OFFER_FINISH_DTTM'],
    ↳ infer_datetime_format=True)
```

```
[4]: len(df_vendas_relampago)
```

```
[4]: 48746
```

```
[5]: values = {"SOLD_AMOUNT": 0, "SOLD_QUANTITY": 0}
df_vendas_relampago.fillna(value=values, inplace=True)
```

```
[6]: min(df_vendas_relampago.OFFER_START_DATE), max(df_vendas_relampago.
    ↳ OFFER_START_DATE)
```

```
[6]: (Timestamp('2021-06-01 00:00:00'), Timestamp('2021-07-31 00:00:00'))
```

```
[7]: df_vendas_relampago.head(2)
```

```
[7]:  OFFER_START_DATE      OFFER_START_DTTM      OFFER_FINISH_DTTM  \
0      2021-06-22  2021-06-22 16:00:00+00:00  2021-06-22 23:02:43+00:00
1      2021-06-22  2021-06-22 13:00:00+00:00  2021-06-22 19:00:02+00:00

      OFFER_TYPE  INVOLVED_STOCK  REMAINING_STOCK_AFTER_END  SOLD_AMOUNT  \
0  lightning_deal              4                      -2         4.72
1  lightning_deal              5                       5         0.00

      SOLD_QUANTITY  ORIGIN  SHIPPING_PAYMENT_TYPE  DOM_DOMAIN_AGG1  VERTICAL  \
0              6.0      A              none      PETS FOOD      CPG
1              0.0    NaN      free_shipping      PET PRODUCTS  OTHERS

      DOMAIN_ID
0      MLM-BIRD_FOODS
1  MLM-ANIMAL_AND_PET_PRODUCTS
```

### 3.2.1 - Extração e preparação da base de feriados

```
[8]: csv_feriados = "datasets/dataset_datas_comem_feriados.csv"

if ~(os.path.exists(csv_feriados)):
```

```
# Extrair base de feriados de junho/2021 caso a base não exista
holidays = ext.GetHolidays()
holidays.get_holidays()
```

```
[9]: df_feriados = pd.read_csv(csv_feriados, parse_dates=['Date'],
    ↳infer_datetime_format=True)
df_feriados["Date"] = pd.to_datetime(df_feriados['Date'], utc=True)
df_feriados['Date'] = df_feriados['Date'].dt.strftime("%Y-%m-%d")
```

```
[10]: df_feriados = df_feriados[df_feriados['Date'] != '2021-06-21']
```

### 3.2.2 - Criar variáveis analíticas

```
[11]: def classify_hours(date):
    hour = date.hour
    if hour >= 6 and hour < 12:
        return "manhã"
    elif hour >= 12 and hour < 18:
        return "tarde"
    else:
        return "noite"
```

```
[12]: # Obter somente a data
df_vendas_relampago['OFFER_FINISH_DATE'] =
    ↳df_vendas_relampago['OFFER_FINISH_DTTM'].dt.strftime("%Y-%m-%d")
```

```
[13]: # Obter dia da semana
df_vendas_relampago['OFFER_START_WEEKDAY'] =
    ↳df_vendas_relampago['OFFER_START_DATE'].dt.strftime("%A")
df_vendas_relampago['OFFER_START_WEEKDAY_ORDER'] =
    ↳df_vendas_relampago['OFFER_START_DATE'].dt.strftime("%w")
```

```
[14]: ## Classificar a hora em manhã (morning), tarde (afternoon) e noite (night)
df_vendas_relampago["OFFER_START_PERIOD"] =
    ↳df_vendas_relampago["OFFER_START_DTTM"].apply(classify_hours)
df_vendas_relampago["OFFER_START_PERIOD_ORDER"] = np.
    ↳where(df_vendas_relampago["OFFER_START_PERIOD"] == 'manhã', 0 ,np.
    ↳where(df_vendas_relampago["OFFER_START_PERIOD"] == 'tarde', 1, 2))
```

```
[15]: # Tratar datas para considerar somente data/hora e minutos
df_vendas_relampago['OFFER_START_DTTM'] =
    ↳df_vendas_relampago['OFFER_START_DTTM'].dt.strftime("%Y-%m-%d %H:%M:%S")
```

```
[16]: # verificar se a promoção ocorreu no mesmo dia somente
df_vendas_relampago['OFFER_SAME_DAY'] = np.where(pd.
    ↳to_datetime(df_vendas_relampago['OFFER_START_DATE']) == pd.
    ↳to_datetime(df_vendas_relampago['OFFER_FINISH_DATE']),1,0)
```

```
[17]: # Verificar se a data de início ou o fim é em um feriado, caso seja em um
      ↪feriado, obter o nome desse feriado também
df_feriados["Date"] = pd.to_datetime(df_feriados['Date'])
df_vendas_relampago = df_vendas_relampago.merge(df_feriados,
      ↪left_on='OFFER_START_DATE', right_on='Date', how="left")
df_vendas_relampago.rename(columns={"Name": "OFFER_START_HOLIDAY", "Date":
      ↪"OFFER_START_HOLIDAY_DATE"}, errors="raise", inplace=True)

df_vendas_relampago["OFFER_FINISH_DATE"] = pd.
      ↪to_datetime(df_vendas_relampago['OFFER_FINISH_DATE'])
df_vendas_relampago = df_vendas_relampago.merge(df_feriados,
      ↪left_on='OFFER_FINISH_DATE', right_on='Date', how="left")
df_vendas_relampago.rename(columns={"Name": "OFFER_FINISH_HOLIDAY", "Date":
      ↪"OFFER_FINISH_HOLIDAY_DATE"}, errors="raise", inplace=True)
```

```
[18]: df_vendas_relampago[df_vendas_relampago['OFFER_START_HOLIDAY'].notnull()][:3]
```

```
[18]:
```

	OFFER_START_DATE	OFFER_START_DTTM	OFFER_FINISH_DTTM	\
2795	2021-06-12	2021-06-12 13:00:00	2021-06-12 16:16:24+00:00	
2796	2021-06-12	2021-06-12 07:00:00	2021-06-12 11:37:44+00:00	
2797	2021-06-12	2021-06-12 19:00:00	2021-06-13 01:00:03+00:00	

	OFFER_TYPE	INVOLVED_STOCK	REMAINING_STOCK_AFTER_END	SOLD_AMOUNT	\
2795	lightning_deal	15	-7	54.23	
2796	lightning_deal	15	0	31.15	
2797	lightning_deal	15	15	0.00	

	SOLD_QUANTITY	ORIGIN	SHIPPING_PAYMENT_TYPE	...	OFFER_FINISH_DATE	\
2795	22.0	NaN	none	...	2021-06-12	
2796	15.0	NaN	none	...	2021-06-12	
2797	0.0	NaN	none	...	2021-06-13	

	OFFER_START_WEEKDAY	OFFER_START_WEEKDAY_ORDER	OFFER_START_PERIOD	\
2795	sábado	6	tarde	
2796	sábado	6	manhã	
2797	sábado	6	noite	

	OFFER_START_PERIOD_ORDER	OFFER_SAME_DAY	OFFER_START_HOLIDAY	\
2795	1	1	Brazilian Valentine's Day	
2796	0	1	Brazilian Valentine's Day	
2797	2	0	Brazilian Valentine's Day	

	OFFER_START_HOLIDAY_DATE	OFFER_FINISH_HOLIDAY	\
2795	2021-06-12	Brazilian Valentine's Day	
2796	2021-06-12	Brazilian Valentine's Day	
2797	2021-06-12	NaN	

	OFFER_FINISH_HOLIDAY_DATE
2795	2021-06-12
2796	2021-06-12
2797	NaT

[3 rows x 23 columns]

## 3 4) Exploração dos dados

### 3.1 4.1 - Gráficos (Visão Geral)

```
[19]: def format_currency(s):
      s = locale.currency(s, grouping=True)
      return s
```

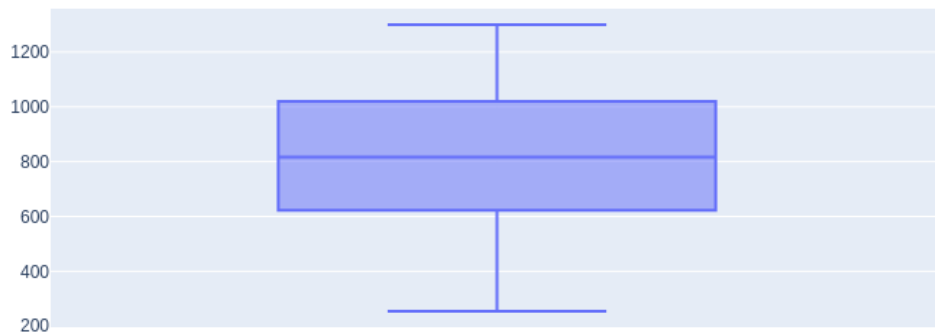
Base sumarizada de ofertas por dia:

```
[20]: df_ofertas = df_vendas_relampago.groupby(['OFFER_START_DATE']).agg({
      "SOLD_QUANTITY": sum,
      "SOLD_AMOUNT": sum,
      "INVOLVED_STOCK": sum,
      "OFFER_START_DTTM": "count",
      "OFFER_START_HOLIDAY": max,
      "OFFER_START_WEEKDAY": max,
      "OFFER_START_WEEKDAY_ORDER": max
    }).reset_index()
df_ofertas.rename(columns={"OFFER_START_DTTM": "QTD_OFFERS"}, inplace=True)
df_ofertas['MEDIAN'] = df_ofertas['QTD_OFFERS'].median()
df_ofertas['MEAN'] = df_ofertas['QTD_OFFERS'].mean()
df_ofertas['MEDIAN_AMOUNT'] = df_ofertas['SOLD_AMOUNT'].median()
df_ofertas['SOLD_AMOUNT_FORM'] = df_ofertas['SOLD_AMOUNT'].
    ↪ apply(format_currency)
df_ofertas['MEDIAN_SOLD_QTD'] = df_ofertas['SOLD_QUANTITY'].median()
df_ofertas['START_HOLIDAY'] = np.where(df_ofertas['OFFER_START_HOLIDAY'].
    ↪ notnull(),1,0)
```

- Distribuição do total de ofertas relâmpago:

```
[21]: fig = px.box(df_ofertas, y="QTD_OFFERS",
      title='Distribuição do total de ofertas relâmpago',
      labels={
          'QTD_OFFERS': "Total de ofertas",
          'OFFER_START_DATE': "Data de início"
      })
fig.update_layout(height=400, width=400, yaxis=None)
fig.show()
```

Distribuição do total de ofertas relâmpago



Nesse gráfico temos a distribuição da quantidade de produtos em promoção relâmpago, com a quantidade de promoções diárias variando de 254 a 1300 por dia e dá para verificar que a mediana separa muito bem esses dados (acima de 816 e abaixo de 816 promoções por dia).

```
[22]: len(df_ofertas[df_ofertas["QTD_OFFERS"] >= 816]),  
      ↪ len(df_ofertas[df_ofertas["QTD_OFFERS"] < 816])
```

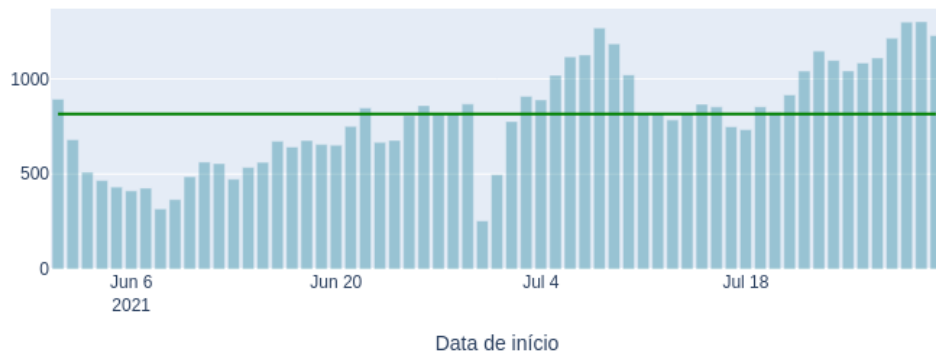
[22]: (31, 30)

- Quantidade de ofertas relâmpago por dia

```
[23]: fig = px.bar(df_ofertas, x='OFFER_START_DATE', y='QTD_OFFERS',  
                  title='Ofertas relâmpago por dia',  
                  labels={  
                      'QTD_OFFERS': "Total de ofertas",  
                      'OFFER_START_DATE': "Data de início"  
                  },  
                  color_discrete_sequence=px.colors.carto.Earth_r,  
                  hover_data=["OFFER_START_DATE", 'QTD_OFFERS'],  
                  opacity=0.4  
                  )  
  
fig.update_layout(yaxis_title=None )  
l = px.line(  
    df_ofertas, x="OFFER_START_DATE", y="MEDIAN"  
).update_traces(line_color="green")  
  
fig.add_traces(l.data)
```

```
fig.show()
```

Ofertas relâmpago por dia



```
[24]: print(df_ofertas.MEDIAN[0]), print(df_ofertas.MEAN[0]);
```

816.0

799.1147540983607

Nesse gráfico podemos observar a quantidade diária de ofertas em relação a mediana e podemos acompanhar melhor em quais dias tivemos mais produtos com ofertas relâmpago. A base encontra-se bem distribuída com a mediana e média muito próximas. Temos então em média 816 produtos diferentes em promoção relâmpago por dia nesse período analisado (01/06/2021 a 31/07/2021)

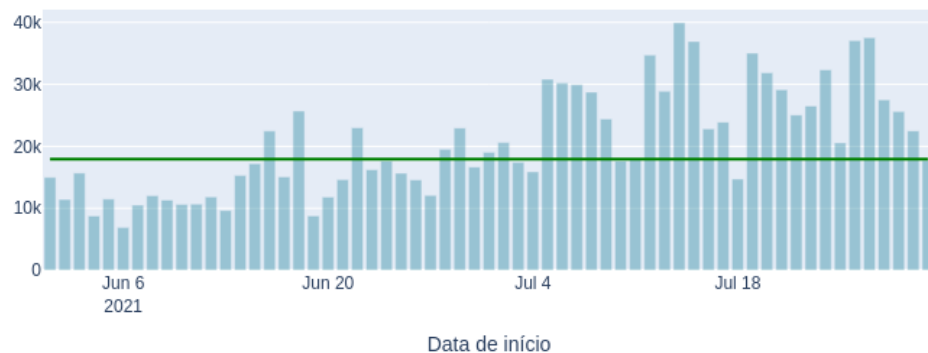
- Valor total diário x valor médio de vendas

```
[25]: df_ofertas['MEDIAN_AMOUNT'] = df_ofertas['SOLD_AMOUNT'].median()
fig = px.bar(df_ofertas,
             x='OFFER_START_DATE', y='SOLD_AMOUNT',
             title='Total em R$ de ofertas relâmpago por dia',
             labels={
                 'SOLD_AMOUNT': "Total de vendas (Em R$)",
                 'OFFER_START_DATE': "Data de início"
             },
             color_discrete_sequence=px.colors.carto.Earth_r,
             hover_data=["OFFER_START_DATE", 'SOLD_AMOUNT'],
             opacity=0.4,
             )
fig.update_layout(yaxis_title=None)
l = px.line(
    df_ofertas, x="OFFER_START_DATE", y="MEDIAN_AMOUNT"
).update_traces(line_color="green")
```

```
fig.add_traces(l.data)

fig.show()
```

Total em R\$ de ofertas relâmpago por dia



Nesse gráfico podemos observar o valor total vendido em relação a mediana e podemos acompanhar melhor em quais dias tivemos boas vendas em comparação ao valor mediano. Temos então, no período analisado um valor médio de R\$ 17.940,64 em vendas de produtos em promoção relâmpago.

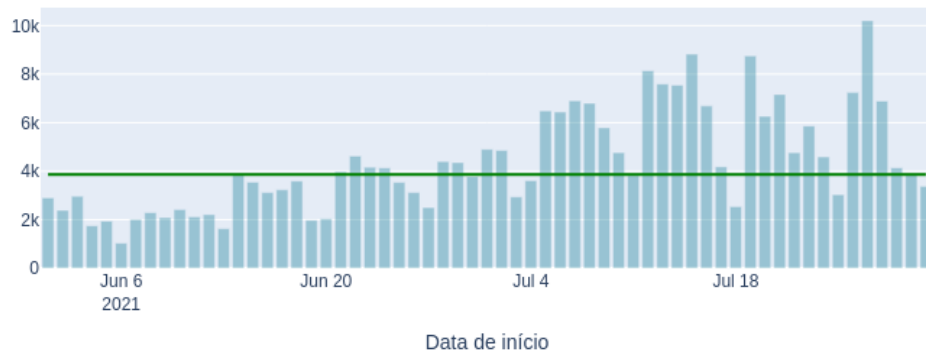
- Quantidade de vendas por dia

```
[26]: df_ofertas['MEDIAN_SOLD_QTD'] = df_ofertas['SOLD_QUANTITY'].median()
fig = px.bar(df_ofertas,
             x='OFFER_START_DATE', y='SOLD_QUANTITY',
             title='Total de produtos vendidos em ofertas relâmpago por dia',
             labels={
                 'SOLD_QUANTITY': "Quantidade total de produtos vendidos por dia",
                 'OFFER_START_DATE': "Data de início"
             },
             color_discrete_sequence=px.colors.carto.Earth_r,
             hover_data=["OFFER_START_DATE", 'SOLD_QUANTITY'],
             opacity=0.4,
         )
fig.update_layout(yaxis_title=None)
l = px.line(
    df_ofertas, x="OFFER_START_DATE", y="MEDIAN_SOLD_QTD"
).update_traces(line_color="green")
fig.add_traces(l.data)
```



```
fig.show()
```

Total de produtos vendidos em ofertas relâmpago por dia



Nesse gráfico podemos observar a quantidade de produtos vendidos em relação a mediana e podemos acompanhar melhor em quais dias tivemos boas vendas em comparação ao valor mediano. Temos então, no período analisado uma quantidade média de 3865 produtos vendidos diariamente em vendas de produtos em promoção relâmpago nesse período.

- Base agrupando por dia da semana

```
[27]: weekday_df = df_ofertas.groupby(["OFFER_START_WEEKDAY",
    ↳ "OFFER_START_WEEKDAY_ORDER"]).agg({
        'SOLD_QUANTITY': sum,
        'SOLD_AMOUNT': sum,
        'INVOLVED_STOCK': sum,
        'QTD_OFFERS': sum
    }).reset_index()
weekday_df['SOLD_AMOUNT_FORM'] = weekday_df['SOLD_AMOUNT'].
    ↳ apply(format_currency)
weekday_df.sort_values(by=['OFFER_START_WEEKDAY_ORDER'], inplace=True,
    ↳ ascending=True)
```

```
[28]: weekday_df[:2]
```

```
[28]:  OFFER_START_WEEKDAY  OFFER_START_WEEKDAY_ORDER  SOLD_QUANTITY  SOLD_AMOUNT  \
0                domingo                        0         20220.0    109591.57
3                segunda                        1         44868.0    197662.01

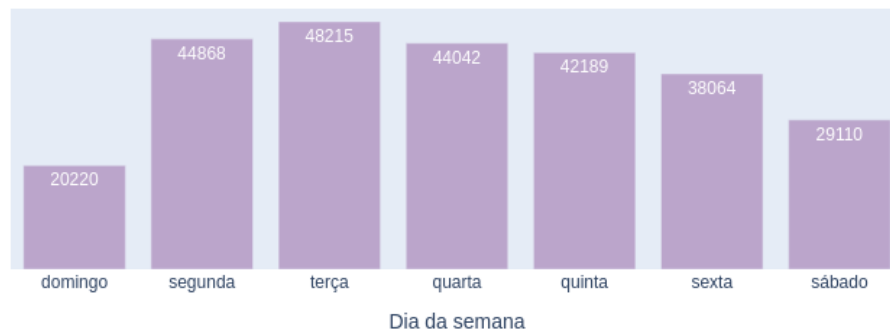
    INVOLVED_STOCK  QTD_OFFERS  SOLD_AMOUNT_FORM
0             151161         5834    R$ 109.591,57
```

3                      266781                      6307                      R\$ 197.662,01

- Quantidade de produtos vendidos por dia da semana

```
[29]: fig = px.bar(weekday_df, x = "OFFER_START_WEEKDAY",
                y = "SOLD_QUANTITY",
                title='Total de unidades de produtos vendidos em ofertas relâmpago_
                por dia da semana',
                labels={
                    'OFFER_START_WEEKDAY': "Dia da semana",
                    'SOLD_QUANTITY': "Total de produtos vendidos"
                },
                color_discrete_sequence=px.colors.carto.Bold,
                hover_data=["OFFER_START_WEEKDAY", 'SOLD_QUANTITY'],
                opacity=0.4,
                text = 'SOLD_QUANTITY'
            )
fig.update_layout(yaxis_title=None, yaxis = None, hovermode=False)
fig.update_yaxes(visible=False)
fig.show()
```

Total de unidades de produtos vendidos em ofertas relâmpago por dia da semana



De forma geral a distribuição de vendas é bem parecida durante os dias com dias em que as vendas são um pouco melhores como segunda e terça e dias em que as vendas são bem menores como sábados e domingos

- Total (em R\$) de produtos vendidos por dia da semana

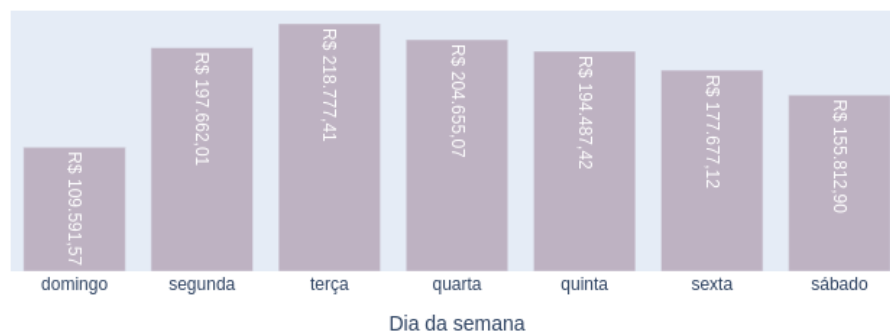
```
[30]: fig = px.bar(weekday_df, x = "OFFER_START_WEEKDAY",
                y = "SOLD_AMOUNT",
```

```

        title='Total de unidades de produtos vendidos em ofertas relâmpago_
↳por dia da semana',
        labels={
            'OFFER_START_WEEKDAY': "Dia da semana",
            'SOLD_AMOUNT': "Total de produtos vendidos"
        },
        color_discrete_sequence=px.colors.carto.Antique,
        hover_data=["OFFER_START_WEEKDAY", 'SOLD_AMOUNT_FORM'],
        opacity=0.4,
        text = 'SOLD_AMOUNT_FORM'
    )
fig.update_layout(yaxis_title=None, hovermode=False)
fig.update_yaxes(visible=False)
fig.show()

```

Total de unidades de produtos vendidos em ofertas relâmpago por dia da semana



- Quantidade de produtos vendidos em feriados x dias comuns

```

[31]: df_ofertas['START_HOLIDAY'] = np.where(df_ofertas['OFFER_START_HOLIDAY'].
↳notnull(),1,0)

df_feriados = df_ofertas.groupby('START_HOLIDAY').agg({'SOLD_QUANTITY':sum,
                                                         "SOLD_AMOUNT":sum,
                                                         "INVOLVED_STOCK":
↳sum,
                                                         "QTD_OFFERS":sum
                                                         }).reset_index()

df_feriados['SOLD_AMOUNT_FORM'] = df_feriados['SOLD_AMOUNT'].
↳apply(format_currency)

```

```
[32]: df_feriados
```

```
[32]:
```

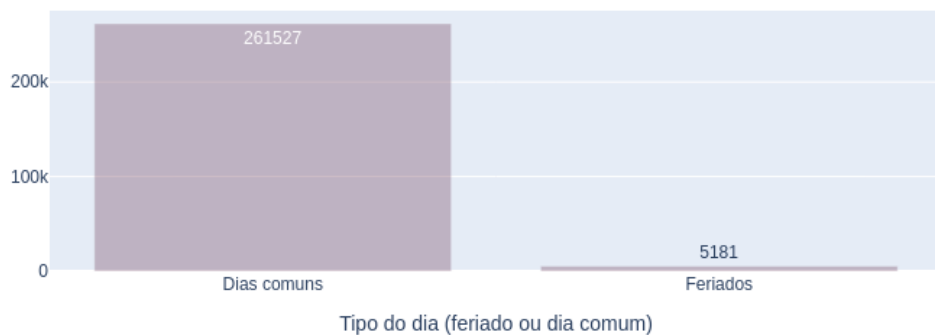
	START_HOLIDAY	SOLD_QUANTITY	SOLD_AMOUNT	INVOLVED_STOCK	QTD_OFFERS \
0	0	261527.0	1231123.26	1668861	47681
1	1	5181.0	27540.24	37615	1065

	SOLD_AMOUNT_FORM
0	R\$ 1.231.123,26
1	R\$ 27.540,24

```
[33]: fig = px.bar(df_feriados, x = "START_HOLIDAY",
                  y = "SOLD_QUANTITY",
                  title='Total de unidades de produtos vendidos em ofertas relâmpago
por dias comuns X feriados',
                  labels={
                      'START_HOLIDAY': "Tipo do dia (feriado ou dia comum)",
                      'SOLD_QUANTITY': "Total de produtos vendidos"
                  },
                  color_discrete_sequence=px.colors.carto.Antique,
                  hover_data=["START_HOLIDAY", 'SOLD_QUANTITY'],
                  opacity=0.4,
                  text = 'SOLD_QUANTITY'
            )
fig.update_layout(yaxis_title=None, hovermode=False)
fig.update_xaxes(type='category')
fig.update_xaxes(
    ticktext=["Dias comuns", "Feriados"],
    tickvals=["0", "1"],
)
fig.show()
```

Total de unidades de produtos vendidos em ofertas relâmpago por dias comuns X feriados



```
[34]: total = df_feriados.SOLD_QUANTITY.sum()
df_feriados.SOLD_QUANTITY / total * 100
```

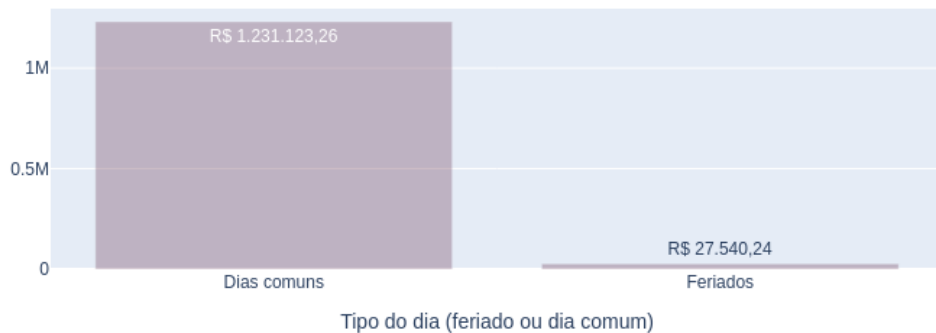
```
[34]: 0    98.057426
      1     1.942574
      Name: SOLD_QUANTITY, dtype: float64
```

Nessa base, a quantidade de produtos vendida em vendas relâmpago em feriados (9164) só corresponde a 3.43% do total de produtos vendidos (266708 produtos)

- Total (em R\$) de produtos vendidos em feriados x dias comuns

```
[35]: fig = px.bar(df_feriados, x = "START_HOLIDAY",
                  y = "SOLD_AMOUNT",
                  title='Total de unidades de produtos vendidos em ofertas relâmpago_
por dias comuns X feriados',
                  labels={
                      'START_HOLIDAY': "Tipo do dia (feriado ou dia comum)",
                      'SOLD_AMOUNT': "Total de produtos vendidos"
                  },
                  color_discrete_sequence=px.colors.carto.Antique,
                  hover_data=["START_HOLIDAY", 'SOLD_AMOUNT'],
                  opacity=0.4,
                  text = 'SOLD_AMOUNT_FORM'
            )
fig.update_layout(yaxis_title=None, hovermode=False)
fig.update_xaxes(type='category')
fig.update_xaxes(
    ticktext=["Dias comuns", "Feriados"],
    tickvals=["0", "1"],
)
fig.show()
```

Total de unidades de produtos vendidos em ofertas relâmpago por dias comuns X feriados



```
[36]: total = df_feriados.SOLD_AMOUNT.sum()
df_feriados.SOLD_AMOUNT / total * 100
```

```
[36]: 0    97.811946
      1     2.188054
      Name: SOLD_AMOUNT, dtype: float64
```

Observa-se na base que o valor total vendido em vendas relâmpago em feriados: 42.177,40, corresponde a 3.35% do faturamento total no período (1.258.663,50)

- Base de dados agrupada por período do dia

```
[37]: df_period = df_vendas_relampago.
      ↳groupby(['OFFER_START_PERIOD', 'OFFER_START_PERIOD_ORDER']).
      ↳agg({'SOLD_QUANTITY':sum,
      "SOLD_AMOUNT":sum,
      "INVOLVED_STOCK":
      ↳sum,
      "OFFER_START_DTTM":
      ↳"count"
      }).reset_index()
df_period['SOLD_AMOUNT_FORM'] = df_period['SOLD_AMOUNT'].apply(format_currency)
df_period.rename(columns={"OFFER_START_DTTM": "QTD_OFFERS"}, inplace=True)
df_period.sort_values(by=['OFFER_START_PERIOD_ORDER'], inplace=True,
      ↳ascending=True)
```

```
[38]: df_period[:2]
```

```
[38]:
```

	OFFER_START_PERIOD	OFFER_START_PERIOD_ORDER	SOLD_QUANTITY	SOLD_AMOUNT	\
0	manhã	0	64109.0	259527.15	
2	tarde	1	154091.0	772359.71	

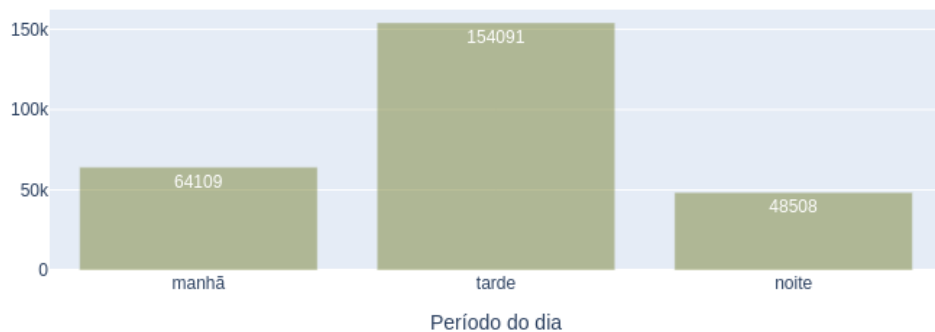
  

	INVOLVED_STOCK	QTD_OFFERS	SOLD_AMOUNT_FORM
0	380922	14376	R\$ 259.527,15
2	1110984	20910	R\$ 772.359,71

- Quantidade de produtos vendidos por turno do dia

```
[39]: fig = px.bar(df_period, x = "OFFER_START_PERIOD_ORDER",
                  y = "SOLD_QUANTITY",
                  title='Total de unidades de produtos vendidos em ofertas relâmpago por período do dia',
                  labels={
                      'OFFER_START_PERIOD_ORDER': "Período do dia",
                      'SOLD_QUANTITY': "Total de produtos vendidos"
                  },
                  color_discrete_sequence=px.colors.carto.Armyrose,
                  hover_data=["OFFER_START_PERIOD", 'SOLD_QUANTITY'],
                  opacity=0.5,
                  text = 'SOLD_QUANTITY'
            )
fig.update_layout(yaxis_title=None, hovermode=False)
fig.update_xaxes(type='category')
fig.update_xaxes(
    ticktext=df_period['OFFER_START_PERIOD'],
    tickvals=["0", "1", "2"],
)
fig.show()
```

Total de unidades de produtos vendidos em ofertas relâmpago por período do dia



```
[40]: total = df_period.SOLD_QUANTITY.sum()
df_period.SOLD_QUANTITY / total * 100
```

```
[40]: 0    24.037149
      2    57.775170
      1    18.187681
      Name: SOLD_QUANTITY, dtype: float64
```

Verifica-se uma tendência de vendas maiores nesse dataset para o período da tarde (cerca de 58% do total de vendas), seguido pelo período da manhã e o período da noite.

- Total de produtos vendidos por turno do dia (em R\$)

```
[41]: fig = px.bar(df_period, x = "OFFER_START_PERIOD_ORDER",
                  y = "SOLD_AMOUNT",
                  title='Total (em R$) de unidades de produtos vendidos em ofertas_
↪relâmpago por período do dia',
                  labels={
                      'OFFER_START_PERIOD_ORDER': "Período do dia",
                      'SOLD_AMOUNT': "Total (em R$) de produtos vendidos"
                  },
                  color_discrete_sequence=px.colors.carto.Earth,
                  hover_data=["OFFER_START_PERIOD", 'SOLD_AMOUNT'],
                  opacity=0.5,
                  text = 'SOLD_AMOUNT_FORM'
            )
fig.update_layout(yaxis_title=None, hovermode=False)
fig.update_xaxes(type='category')
fig.update_xaxes(
    ticktext=df_period['OFFER_START_PERIOD'],
    tickvals=["0", "1", "2"],
)
fig.show()
```

Total (em R\$) de unidades de produtos vendidos em ofertas relâmpago por período do dia





```
[42]: total = df_period.SOLD_AMOUNT.sum()
df_period.SOLD_AMOUNT / total * 100
```

```
[42]: 0    20.619264
      2    61.363479
      1    18.017257
      Name: SOLD_AMOUNT, dtype: float64
```

Esses 58% do total de vendas relâmpago a tarde são responsáveis por 61% do valor total vendido (em R\$) no período analisado em vendas relâmpago no período da tarde. 61% do valor de 1.258.663,50

De uma forma geral, há um insight na base relacionado ao turno em que a oferta relâmpago é realizada. No período da tarde temos grande partes das vendas, sendo também a maior parte do valor total de vendas

### 3.1.1 Vendas por tipo de frete

- Base agrupada por tipo de frete

```
[43]: df_frete = df_vendas_relampago.groupby('SHIPPING_PAYMENT_TYPE').agg({
      "SOLD_QUANTITY":sum,
      "SOLD_AMOUNT":sum,
      "INVOLVED_STOCK":sum,
      "OFFER_START_DTTM":"count"
    }).reset_index()
df_frete.rename(columns={"OFFER_START_DTTM": "QTD_OFFERS"}, inplace=True)
df_frete['SHIPPING_PAYMENT_TYPE'] = np.where(df_frete['SHIPPING_PAYMENT_TYPE']_
↪ == 'none', 'pago_cliente', 'free_shipping')
```

```
[44]: df_frete
```

```
[44]:  SHIPPING_PAYMENT_TYPE  SOLD_QUANTITY  SOLD_AMOUNT  INVOLVED_STOCK  \
0         free_shipping         64347.0      748536.12         592537
1         pago_cliente         202361.0      510127.38         1113939

      QTD_OFFERS
0           26658
1           22088
```

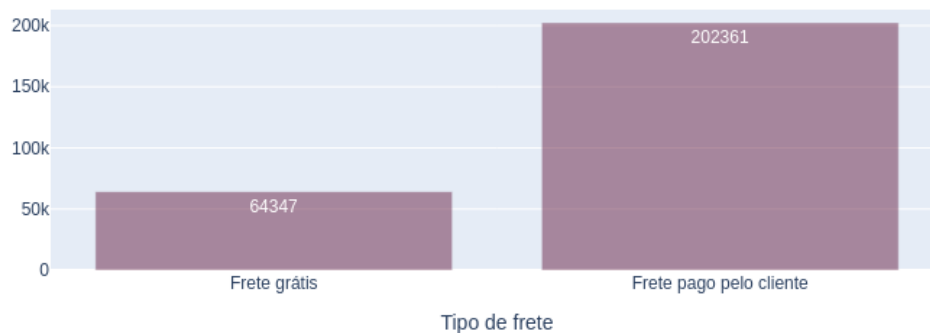
```
[45]: fig = px.bar(df_frete, x = "SHIPPING_PAYMENT_TYPE",
      y = "SOLD_AMOUNT",
      title='Total de unidades de produtos vendidos em ofertas relâmpago_
↪por período do dia',
      labels={
        'SHIPPING_PAYMENT_TYPE': "Tipo de frete",
        'SOLD_AMOUNT': "Total de produtos vendidos"
      },
```

```

        color_discrete_sequence=px.colors.carto.Burg_r,
        hover_data=["SHIPPING_PAYMENT_TYPE", 'SOLD_QUANTITY'],
        opacity=0.5,
        text = 'SOLD_QUANTITY'
    )
    fig.update_layout(yaxis_title=None, hovermode=False)
    fig.update_xaxes(type='category')
    fig.update_xaxes(
        ticktext=["Frete grátis", "Frete pago pelo cliente"],
        tickvals=["0", "1"],
    )
    fig.show()

```

Total de unidades de produtos vendidos em ofertas relâmpago por período do dia



### 3.2 Gráficos por categoria

Vamos verificar agora o comportamento dos dados em relação a categoria dos produtos:

```

[46]: df_ofertas_cat = df_vendas_relampago.groupby(['OFFER_START_DATE', 'VERTICAL']).
      ↪agg({
          "SOLD_QUANTITY":sum,
          "SOLD_AMOUNT":sum,
          "INVOLVED_STOCK":sum,
          "OFFER_START_DTTM":"count"
      }).reset_index()
      df_ofertas_cat.rename(columns={"OFFER_START_DTTM": "QTD_OFFERS"}, inplace=True)

```

```

[47]: df_ofertas_cat[:5]

```

```
[47]:
```

	OFFER_START_DATE	VERTICAL	SOLD_QUANTITY	SOLD_AMOUNT	\
0	2021-06-01	ACC	41.0	220.00	
1	2021-06-01	APP & SPORTS	257.0	1604.70	
2	2021-06-01	BEAUTY & HEALTH	1903.0	5406.73	
3	2021-06-01	CE	275.0	3880.74	
4	2021-06-01	CPG	84.0	389.55	

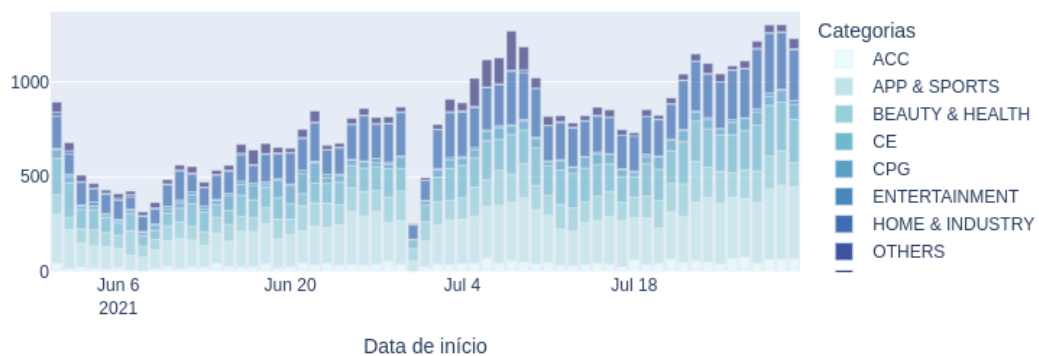
	INVOLVED_STOCK	QTD_OFFERS
0	510	48
1	2381	257
2	14528	100
3	2116	193
4	585	46

### 3.2.1 Quantidade de ofertas por categoria

- Ofertas por data e categoria do produto

```
[48]: fig = px.bar(df_ofertas_cat, x='OFFER_START_DATE', y='QTD_OFFERS',
                  color="VERTICAL", title='Ofertas relâmpago por Categoria',
                  labels={
                      'QTD_OFFERS': "Total de ofertas",
                      "VERTICAL": "Categorias",
                      'OFFER_START_DATE': "Data de início"
                  },
                  hover_data=["VERTICAL", 'OFFER_START_DATE', 'QTD_OFFERS'],
                  color_discrete_sequence=px.colors.sequential.ice_r,
                  opacity=0.7
                  )
fig.update_layout(yaxis_title=None)
fig.show()
```

Ofertas relâmpago por Categoria

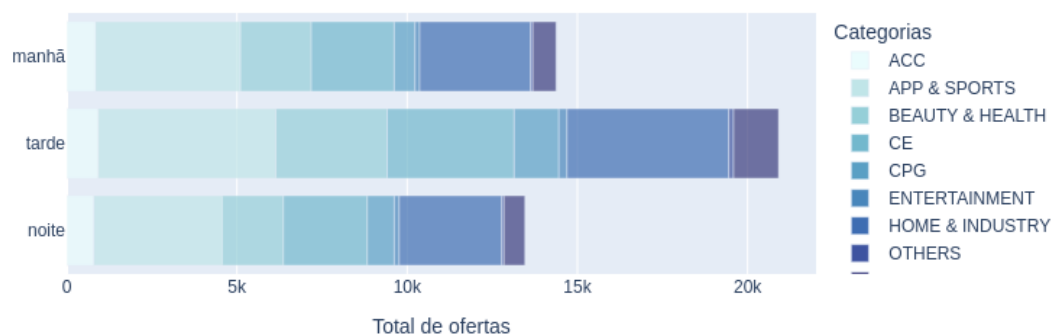


- Turnos de ofertas por categoria

```
[49]: df_turnos_ofertas_cat = df_vendas_relampago.
      ↪groupby(['OFFER_START_PERIOD', 'VERTICAL'])[['OFFER_START_PERIOD_ORDER', 'VERTICAL']].
      ↪value_counts().reset_index()
df_turnos_ofertas_cat.rename(columns={0: "QTDE"}, inplace=True)
df_turnos_ofertas_cat.sort_values(by=['OFFER_START_PERIOD_ORDER', 'VERTICAL'],
      ↪inplace=True, ascending=True)

[50]: fig = px.bar(df_turnos_ofertas_cat, x="QTDE", y="OFFER_START_PERIOD",
      ↪color='VERTICAL', orientation='h',
          title='Turnos de ofertas relâmpago por Categoria',
          labels={
              'QTDE': "Total de ofertas",
              "VERTICAL": "Categorias",
              'OFFER_START_PERIOD': "Turno da oferta relâmpago"
          },
          category_orders={"OFFER_START_PERIOD": ["manhã", "tarde", "noite"]},
          hover_data=["VERTICAL", 'OFFER_START_PERIOD', 'QTDE'],
          color_discrete_sequence=px.colors.sequential.ice_r,
          opacity=0.7,
      )
fig.update_layout(yaxis_title=None)
fig.show()
```

Turnos de ofertas relâmpago por Categoria



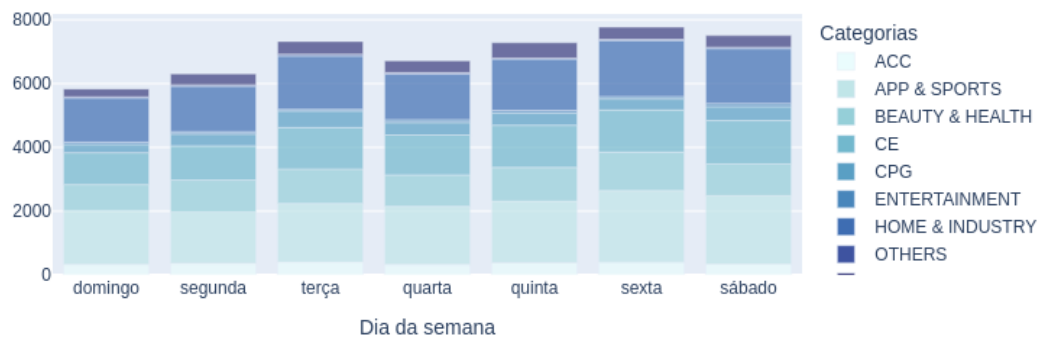
Nesse gráfico podemos verificar que, de forma geral no período da tarde temos um aumento de ofertas em todas as categorias

- Ofertas por dia da semana

```
[51]: df_ofertas_weekday = df_vendas_relampago.
      ↪groupby(['OFFER_START_WEEKDAY', 'VERTICAL'])[['OFFER_START_WEEKDAY_ORDER', 'VERTICAL']].
      ↪value_counts().reset_index()
df_ofertas_weekday.rename(columns={0: "QTDE"}, inplace=True)
df_ofertas_weekday.sort_values(by=['OFFER_START_WEEKDAY_ORDER', 'VERTICAL'],
      ↪inplace=True, ascending=True)

[52]: fig = px.bar(df_ofertas_weekday, x='OFFER_START_WEEKDAY', y='QTDE',
                  color="VERTICAL", title='Ofertas relâmpago por dia da semana',
                  labels={
                      'QTDE': "Total de ofertas",
                      "VERTICAL": "Categorias",
                      'OFFER_START_WEEKDAY': "Dia da semana"
                  },
                  hover_data=["VERTICAL", 'OFFER_START_WEEKDAY', 'QTDE'],
                  color_discrete_sequence=px.colors.sequential.ice_r,
                  opacity=0.7
                  )
fig.update_layout(yaxis_title=None)
fig.show()
```

Ofertas relâmpago por dia da semana



Em relação ao dia da semana, a proporção da quantidade de ofertas em relação as categorias é bem similar, não havendo alterações tão significativas, com os maiores valores na terça e na sexta em geral

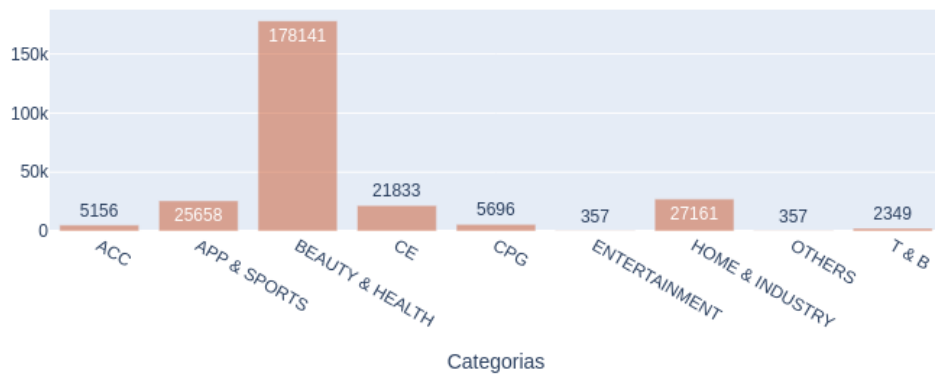
### 3.2.2 Produtos vendidos por categoria

```
[53]: df_ofertas_cat = df_vendas_relampago.groupby(['VERTICAL']).agg({
        'SOLD_QUANTITY':sum,
        'SOLD_AMOUNT':sum
    }).reset_index()
df_ofertas_cat['SOLD_AMOUNT_FORM'] = df_ofertas_cat['SOLD_AMOUNT'].
    ↪apply(format_currency)
df_ofertas_cat.sort_values(by=['VERTICAL'], inplace=True, ascending=True)
```

- Quantidade de produtos por categoria

```
[54]: fig = px.bar(df_ofertas_cat, x='VERTICAL', y='SOLD_QUANTITY',
        title='Produtos vendidos nas ofertas relâmpago por categoria',
        labels={
            'SOLD_QUANTITY': "Total de produtos vendidos",
            "VERTICAL": "Categorias"
        },
        color_discrete_sequence=px.colors.carto.Geyser_r,
        opacity=0.5,
        text = 'SOLD_QUANTITY'
    )
fig.update_layout(yaxis_title=None, hovermode=False)
fig.show()
```

Produtos vendidos nas ofertas relâmpago por categoria



```
[55]: total = df_ofertas_cat.SOLD_QUANTITY.sum()
df_ofertas_cat.SOLD_QUANTITY / total * 100
```

```
[55]: 0    1.933200
      1    9.620259
```

```

2    66.792522
3     8.186106
4    2.135669
5     0.133854
6    10.183797
7     0.133854
8     0.880738
Name: SOLD_QUANTITY, dtype: float64

```

Nas vendas relâmpago, a categoria mais vendida é a de produtos de beleza e saúde (BEAUTY & HEALTH), representando 66,8% do total das vendas, seguida pelas categorias APP & SPORTS (9,62%) e HOME & INDUSTRY (10,18%).

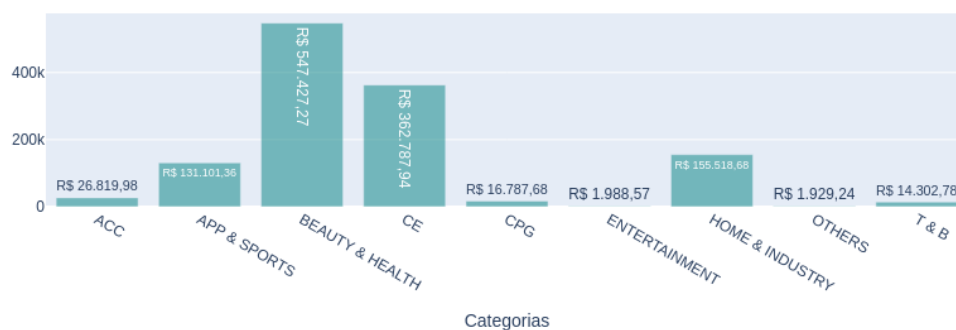
- Valor total de produtos vendidos por categoria

```

[56]: fig = px.bar(df_ofertas_cat, x='VERTICAL', y='SOLD_AMOUNT',
                  title='Produtos vendidos nas ofertas relâmpago por categoria',
                  labels={
                      'SOLD_AMOUNT': "Total (EM R$) em produtos vendidos",
                      "VERTICAL": "Categorias"
                  },
                  color_discrete_sequence=px.colors.carto.Geyser,
                  opacity=0.5,
                  text = 'SOLD_AMOUNT_FORM'
                  )
fig.update_layout(yaxis_title=None, hovermode=False)
fig.show()

```

Produtos vendidos nas ofertas relâmpago por categoria



```

[57]: total = df_ofertas_cat.SOLD_AMOUNT.sum()
df_ofertas_cat.SOLD_AMOUNT / total * 100

```

```
[57]: 0      2.130830
      1     10.415918
      2     43.492742
      3     28.823267
      4      1.333770
      5      0.157991
      6     12.355858
      7      0.153277
      8      1.136347
      Name: SOLD_AMOUNT, dtype: float64
```

Ao observar o valor vendido, verifica-se que os percentuais se encontram um pouco mais distribuídos entre as categorias quando comparamos as quantidades vendidas, mas a categoria de produtos de beleza e saúde (BEAUTY & HEALTH) se mantém em primeiro lugar, representando 43,5% do total em R\$ de vendas realizadas, sendo seguida pela categoria CE (28,82%) e a de HOME & INDUSTRY (12.35%)

## 4 Resumo das análises e insights encontrados

### 4.1 Informações do dataset

A base contém dados de 01/06/2021 a 31/07/2021 e possui 48746 registros

### 4.2 Base em Geral:

- A quantidade de promoções diárias varia de 254 a 1300 por dia e dá para verificar que a mediana separa muito bem esses dados (acima de 816 e abaixo de 816 promoções por dia).
- Temos então em média 816 produtos diferentes em promoção relâmpago por dia nesse período analisado
- No período analisado temos um valor médio de 17.940,64 em vendas de produtos em promoção relâmpago e uma quantidade média de 3865 produtos vendidos diariamente
- De forma geral a distribuição de vendas é bem parecida durante os dias com dias em que as vendas são um pouco melhores como segunda e terça e dias em que as vendas são bem menores como sábados e domingos
- Nessa base, a quantidade de produtos vendida em vendas relâmpago em feriados (9164) só corresponde a 3.43% do total de produtos vendidos (266708 produtos)
- 58% do total de vendas relâmpago são iniciadas a tarde, e são responsáveis por 61% do valor total vendido (em R\$) no período analisado em vendas relâmpago no período (61% do valor de 1.258.663,50)
- De uma forma geral, há um insight na base relacionado ao turno em que a oferta relâmpago é realizada. No período da tarde temos grande partes das vendas, sendo também a maior parte do valor total de vendas.
- Na maior parte de compras desse tipo de ofertas o frete é pago pelo cliente.

### 4.3 Categoria

- De forma geral no período da tarde temos um aumento de ofertas em todas as categorias (turno), sendo algumas mais expressivas no gráfico



- Em relação ao dia da semana, a proporção da quantidade de ofertas em relação as categorias é bem similar, não havendo alterações tão significativas, com os maiores valores na terça e na sexta em geral
- Nas vendas relâmpago, a categoria mais vendida é a de produtos de beleza e saúde (BEAUTY & HEALTH), representando 66,8% do total das vendas, seguida pelas categorias APP & SPORTS (9,62%) e HOME & INDUSTRY (10,18%).
- Ao observar o valor vendido, verifica-se que os percentuais se encontram um pouco mais distribuídos entre as categorias quando comparamos as quantidades vendidas, mas a categoria de produtos de beleza e saúde (BEAUTY & HEALTH) se mantém em primeiro lugar, representando 43,5% do total em R\$ de vendas realizadas, sendo seguida pela categoria CE (28,82%) e a de HOME & INDUSTRY (12,35%)