

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from datetime import datetime, timedelta
from datetime import date
import datetime as dt

sns.set_style()
```

In [2]:

```
%matplotlib inline
```

In [4]:

```
# Importando o dataset da Hamburgueria 207

df= pd.read_excel('Historico_Itens_Vendidos de 01-01-18 à 31-12-18.xls')
```

In [5]:

```
df.head()
```

Out[5]:

	Data/Hora Item	Qtd.	Valor Un. Item	Valor. Tot. Item	Tipo de Item	Nome Prod.	Tipo Prod.	Cat. Prod.	V P
0	2018-03-05 19:03:29.296	1.0	17.0	17.0	Produto	207 +BACON	Produto	Hambúrgueres	
1	2018-03-05 19:06:18.572	1.0	22.0	22.0	Produto	207 PRIME	Produto	Hambúrgueres	
2	2018-03-05 19:06:24.578	1.0	3.0	3.0	Produto	COCA COLA LATA	Produto	Refrigerantes	
3	2018-03-05 19:10:04.523	1.0	15.0	15.0	Produto	207 CLÁSSICO	Produto	Hambúrgueres	
4	2018-03-05 19:10:04.523	1.0	1.5	1.5	Complemento	QUEIJO ADICIONAL	Complemento	Complemento	

In [45]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10532 entries, 0 to 10531
Data columns (total 16 columns):
Data                10532 non-null datetime64[ns]
Qtd                 10532 non-null float64
ValorUnidade       10532 non-null float64
TotalUnidade       10532 non-null float64
Produto            10532 non-null object
TipoProduto        10532 non-null object
Categoria          10532 non-null object
ValorProduto       10532 non-null float64
HoraAbertPed       10532 non-null datetime64[ns]
HoraFechPed        10532 non-null datetime64[ns]
TipoPedido         10532 non-null object
HoraAbertura       10532 non-null object
HoraFechamento    10532 non-null object
DiaSemana          10532 non-null object
Mes                10532 non-null object
Hora               10532 non-null int64
dtypes: datetime64[ns](3), float64(4), int64(1), object(8)
memory usage: 1.3+ MB
```

Tamanho do DataFrame

In [7]:

```
df.shape
```

Out[7]:

```
(10533, 15)
```

Valores missings

In [8]:

```
df.isnull().sum()
```

Out[8]:

```
Data/Hora Item      1
Qtd.                1
Valor Un. Item      1
Valor. Tot. Item    1
Tipo de Item        1
Nome Prod.          1
Tipo Prod.          1
Cat. Prod.          1
Valor Prod.         1
Cod. Ped.           1
Núm. Mesa/Com.     8266
Data Ab. Ped.       1
Data Fec. Ped.      1
Tipo Ped.           1
Stat. Ped.          1
dtype: int64
```

In [9]:

```
# Verificando o %
```

```
df.isnull().sum() / df.shape[0]
```

Out[9]:

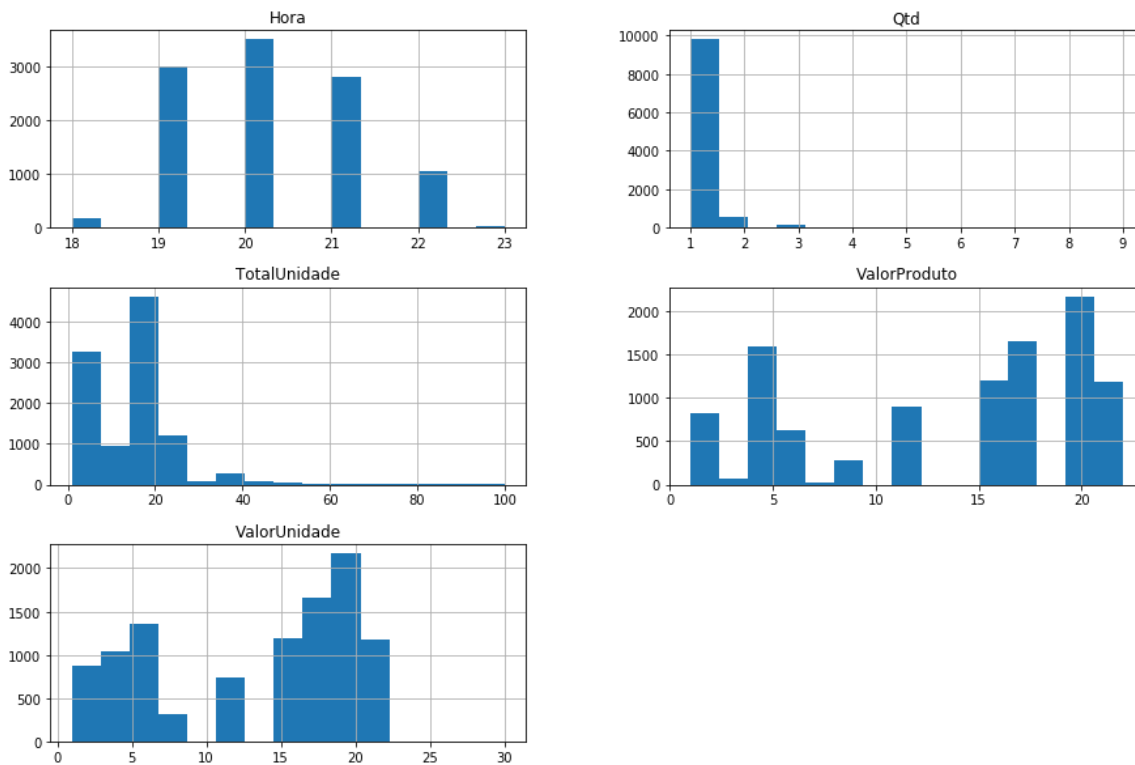
```
Data/Hora Item      0.000095
Qtd.                0.000095
Valor Un. Item      0.000095
Valor. Tot. Item    0.000095
Tipo de Item        0.000095
Nome Prod.          0.000095
Tipo Prod.          0.000095
Cat. Prod.          0.000095
Valor Prod.         0.000095
Cod. Ped.           0.000095
Núm. Mesa/Com.     0.784772
Data Ab. Ped.       0.000095
Data Fec. Ped.      0.000095
Tipo Ped.           0.000095
Stat. Ped.          0.000095
dtype: float64
```

In [46]:

```
df.hist(bins=15, figsize=(15,10))
```

Out[46]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0CD3FF70>,  
      <matplotlib.axes._subplots.AxesSubplot object at 0x0CFB2AF0>],  
      [<matplotlib.axes._subplots.AxesSubplot object at 0x0CFD3BD0>,  
      <matplotlib.axes._subplots.AxesSubplot object at 0x0D033CB0>],  
      [<matplotlib.axes._subplots.AxesSubplot object at 0x0D053D90>,  
      <matplotlib.axes._subplots.AxesSubplot object at 0x0D073E70>]],  
      dtype=object)
```



Medidas de tendência

In [11]:

```
df.describe()
```

Out[11]:

	Qtd.	Valor Un. Item	Valor. Tot. Item	Valor Prod.	Cod. Ped.	Núm. Mesa/Com.
count	10532.000000	10532.000000	10532.000000	10532.000000	10532.000000	2267.000000
mean	1.090011	13.239271	14.525826	13.475978	8922.573111	7.341420
std	0.360771	7.106302	9.562887	6.910087	1503.848371	18.970337
min	1.000000	1.000000	1.000000	1.000000	6187.000000	1.000000
25%	1.000000	5.000000	5.500000	6.000000	7636.000000	1.000000
50%	1.000000	15.000000	15.000000	16.000000	8943.500000	2.000000
75%	1.000000	20.000000	20.000000	20.000000	10287.250000	7.000000
max	9.000000	30.000000	100.000000	22.000000	11369.000000	102.000000

In [12]:

```
# Coluna Numero Mesa / Comanda foi retirada do dataset por ter valores nulos
df.drop(["Núm. Mesa/Com."], axis=1, inplace=True)
```

In [13]:

```
# Coluna Codigo Pedido foi retirada do dataset por ter valores nulos
df.drop(["Cod. Ped."], axis=1, inplace=True)
```

In [14]:

```
# Verificando os nomes das colunas do dataset
df.columns
```

Out[14]:

```
Index(['Data/Hora Item', 'Qtd.', 'Valor Un. Item', 'Valor. Tot. Item',
      'Tipo de Item', 'Nome Prod.', 'Tipo Prod.', 'Cat. Prod.', 'Valor Pr
od.',
      'Data Ab. Ped.', 'Data Fec. Ped.', 'Tipo Ped.', 'Stat. Ped.'],
      dtype='object')
```

In [15]:

```
# Renomeando as colunas do dataset

colunaNova = ['Data', 'Qtd', 'ValorUnidade', 'TotalUnidade',
              'TipoItem', 'Produto', 'TipoProduto', 'Categoria', 'ValorProduto',
              'HoraAbertPed', 'HoraFechPed', 'TipoPedido', 'StatuSPedido']
df.columns = colunaNova
```

In [16]:

Verificando colunas novas

df.columns

Out[16]:

```
Index(['Data', 'Qtd', 'ValorUnidade', 'TotalUnidade', 'TipoItem', 'Produto',
      'TipoProduto', 'Categoria', 'ValorProduto', 'HoraAbertPed',
      'HoraFechPed', 'TipoPedido', 'StatusPedido'],
      dtype='object')
```

In [17]:

```
# Removendo a coluna TipoItem, pois a coluna TipoProduto possui as mesmas informações
df.drop(["TipoItem"], axis=1, inplace=True)
```

In [18]:

Removendo a coluna StatusPedido

df.drop(["StatusPedido"], axis=1, inplace=True)

In [19]:

Verificando a última linha do dataset e encontramos valores missings

df.tail()

Out[19]:

	Data	Qtd	ValorUnidade	TotalUnidade	Produto	TipoProduto	Categoria
10528	2018-12-30 22:37:39.490	1.0	5.0	5.0	* Excluído * SUCO	Produto	Sucos
10529	2018-12-30 22:41:57.277	1.0	2.5	2.5	AGUA MINERAL 500ML	Produto	Refrigerantes
10530	2018-12-30 22:41:59.290	1.0	2.5	2.5	AGUA MINERAL 500ML	Produto	Refrigerantes
10531	2018-12-30 22:47:23.911	1.0	6.5	6.5	COCA COLA 1 LT	Produto	Refrigerantes
10532	NaT	NaN	NaN	NaN	NaN	NaN	NaN

In [20]:

```
# Excluindo a ultima linha do dataset, pois não existem valores
df.drop([10532], axis=0, inplace=True)
```

In [21]:

```
# Nenhum valor nulo no dataset
df.isnull().sum().all
```

Out[21]:

```
<bound method Series.all of Data          0
Qty          0
ValorUnidade  0
TotalUnidade  0
Produto       0
TipoProduto   0
Categoria     0
ValorProduto  0
HoraAbertPed   0
HoraFechPed    0
TipoPedido    0
dtype: int64>
```

In [22]:

```
# Agora vamos criar dias da semana, meses e hora do dia para extrair insights
df.loc[0][0]
```

Out[22]:

```
Timestamp('2018-03-05 19:03:29.296000')
```

In [23]:

```
# colocando a variavel data como date.time
df["Data"] = pd.to_datetime(df["Data"])
```

In [24]:

```
# Criando copia do dataset
hb = df
```

In [25]:

```
# Extraindo e criando as horas das colunas aberturaped e fechamentoped
df["HoraAbertura"] = df["HoraAbertPed"].apply(lambda x:dt.time(x.hour, x.minute, x.second))
df["HoraFechamento"] = df["HoraFechPed"].apply(lambda x:dt.time(x.hour, x.minute, x.second))
```

In [26]:

```
# Extraindo e criando colunas de Dias da Semana e Mês
df["DiaSemana"] = df["Data"].apply(lambda x: x.dayofweek)
df["Mes"] = df["Data"].apply(lambda x: x.month)
```

In [27]:

```
df["Hora"] = df["Data"].apply(lambda x: x.hour)
```

In [29]:

```
# Criando um dicionário com os nomes dos dias da semana

dias = {0:"Seg",1:"Ter",2:"Qua",3:"Qui",4:"Sex",5:"Sab",6:"Dom"}
```

In [30]:

```
# Atribuindo os nomes para a coluna DiaSemana

df["DiaSemana"] = df["DiaSemana"].map(dias)
```

In [31]:

```
# Criando os meses tambem

meses = {1:"Jan",2:"Fev",3:"Mar",4:"Abr",5:"Mai",6:"Jun",7:"Jul",8:'Ago',9:"Set",10:"Out",11:"Nov",12:"Dez"}
```

In [32]:

```
df["Mes"] = df["Mes"].map(meses)
```

In [33]:

```
# Visualizando o dataset com as novas colunas de horas, meses e dias

df.head()
```

Out[33]:

	Data	Qtd	ValorUnidade	TotalUnidade	Produto	TipoProduto	Categoria	V
0	2018-03-05 19:03:29.296	1.0	17.0	17.0	207 +BACON	Produto	Hambúrgueres	
1	2018-03-05 19:06:18.572	1.0	22.0	22.0	207 PRIME	Produto	Hambúrgueres	
2	2018-03-05 19:06:24.578	1.0	3.0	3.0	COCA COLA LATA	Produto	Refrigerantes	
3	2018-03-05 19:10:04.523	1.0	15.0	15.0	207 CLÁSSICO	Produto	Hambúrgueres	
4	2018-03-05 19:10:04.523	1.0	1.5	1.5	QUEIJO ADICIONAL	Complemento	Complemento	

In [34]:

```
df["DiaSemana"].value_counts()
```

Out[34]:

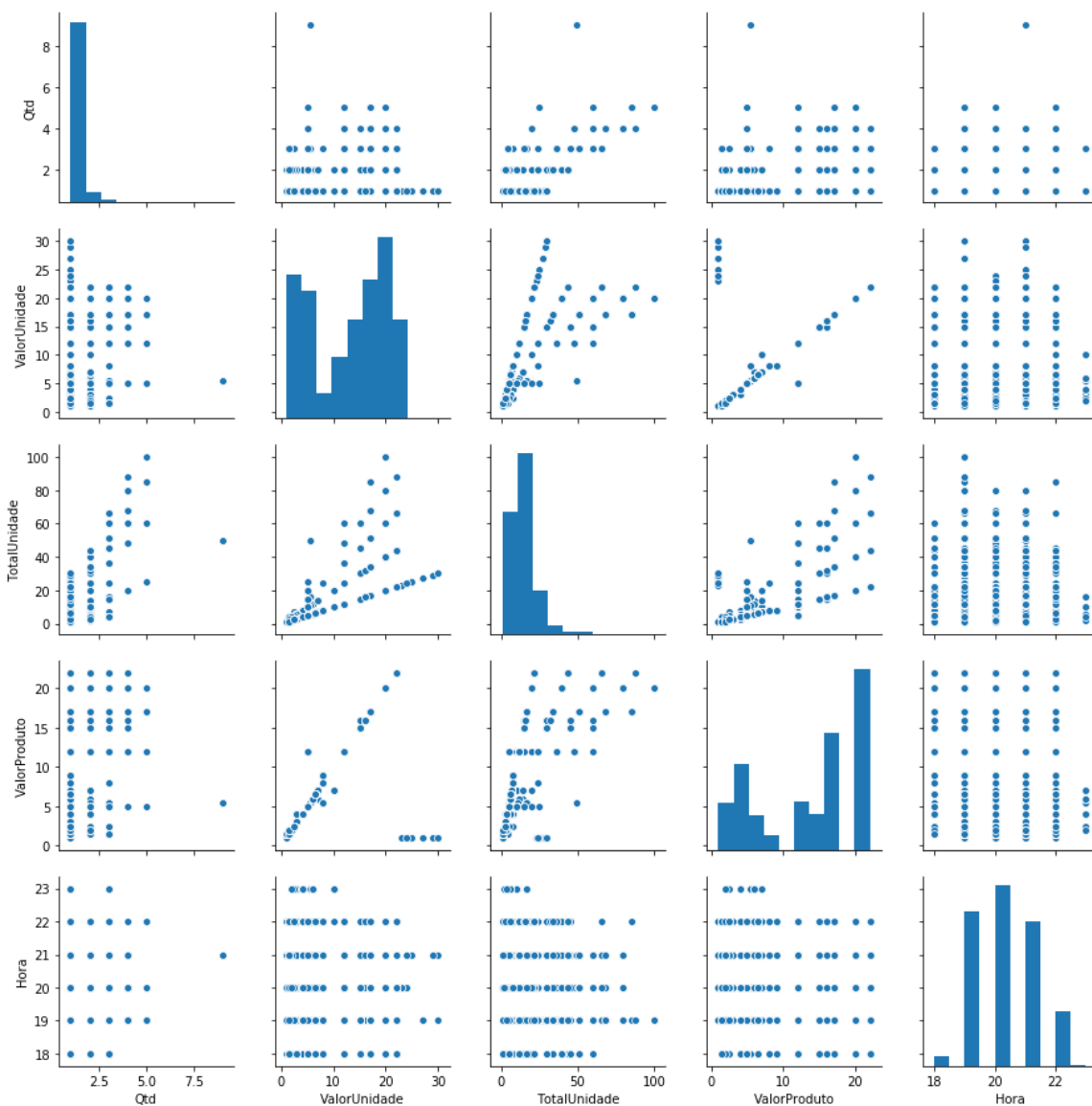
```
Dom    3326
Sex    1642
Sab    1619
Qui    1464
Qua    1401
Seg    1031
Ter      49
Name: DiaSemana, dtype: int64
```

In [35]:

```
sns.pairplot(df)
```

Out[35]:

```
<seaborn.axisgrid.PairGrid at 0xa0c7430>
```



Criando gráficos para análises

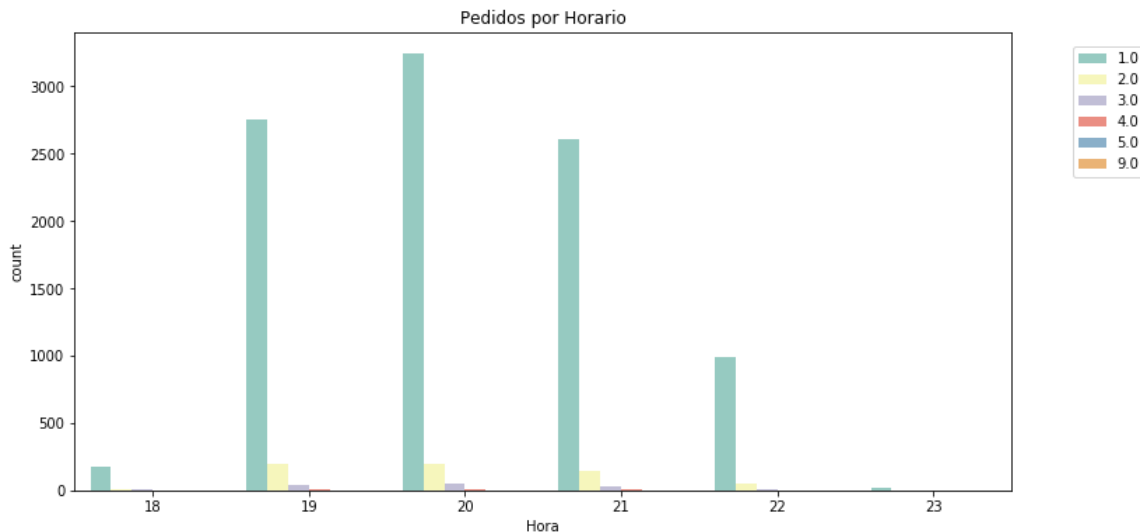
In [36]:

```
# Criando graficos para extrair informações
```

```
plt.figure(figsize=(12,6))  
sns.countplot(x = "Hora", hue= "Qtd", data=df, palette="Set3")  
plt.title("Pedidos por Horário")  
plt.legend(bbox_to_anchor=(1.05,1), loc = 0, borderaxespad=1)
```

Out[36]:

<matplotlib.legend.Legend at 0xb2fff0>

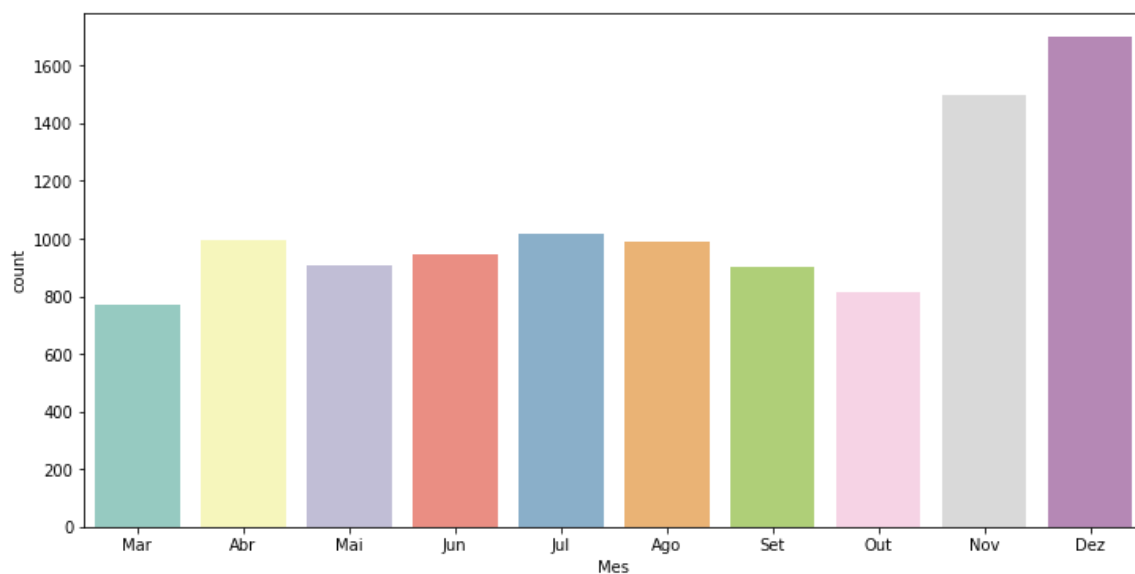


In [37]:

```
plt.figure(figsize=(12,6))  
sns.countplot(x="Mes", data=df, palette="Set3")
```

Out[37]:

<matplotlib.axes._subplots.AxesSubplot at 0xaa06870>

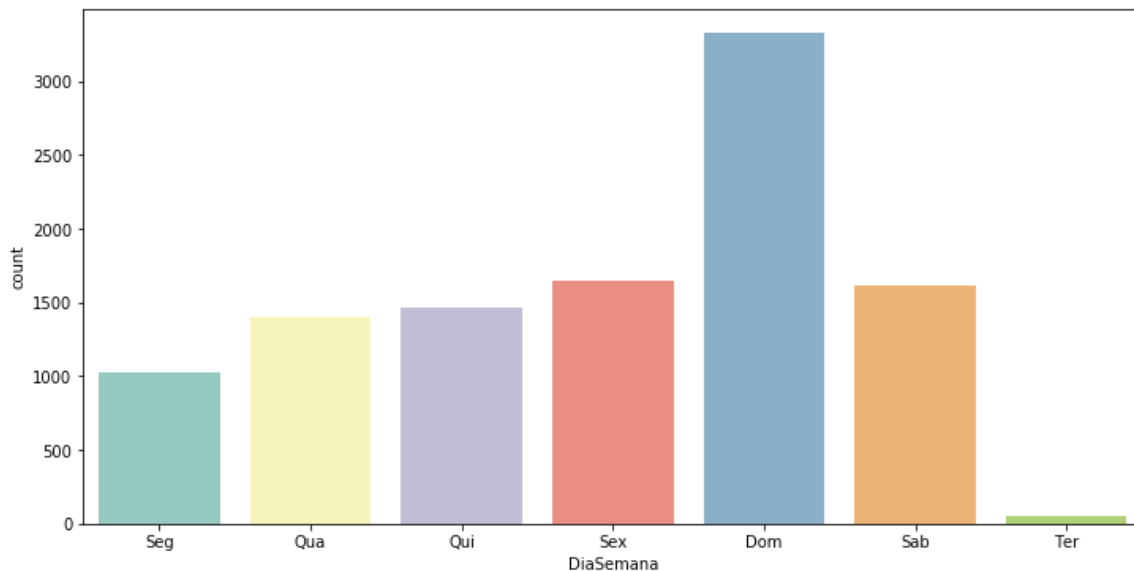


In [38]:

```
plt.figure(figsize=(12,6))
sns.countplot(x= "DiaSemana", data=df, palette="Set3")
```

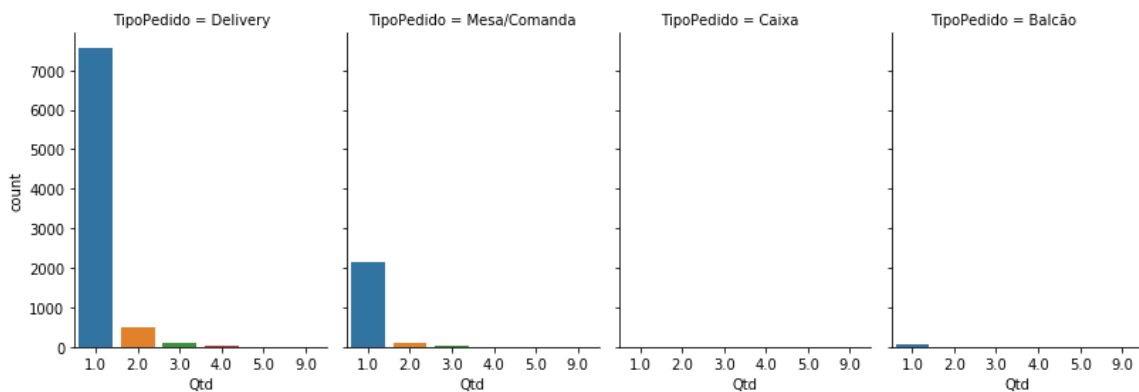
Out[38]:

<matplotlib.axes._subplots.AxesSubplot at 0x4fa4b50>



In [39]:

```
sns.catplot(x="Qtd", col="TipoPedido",
            data=df, kind="count",
            height=4, aspect=.7);
```



In [40]:

```
df["TipoPedido"].value_counts() / 100
```

Out[40]:

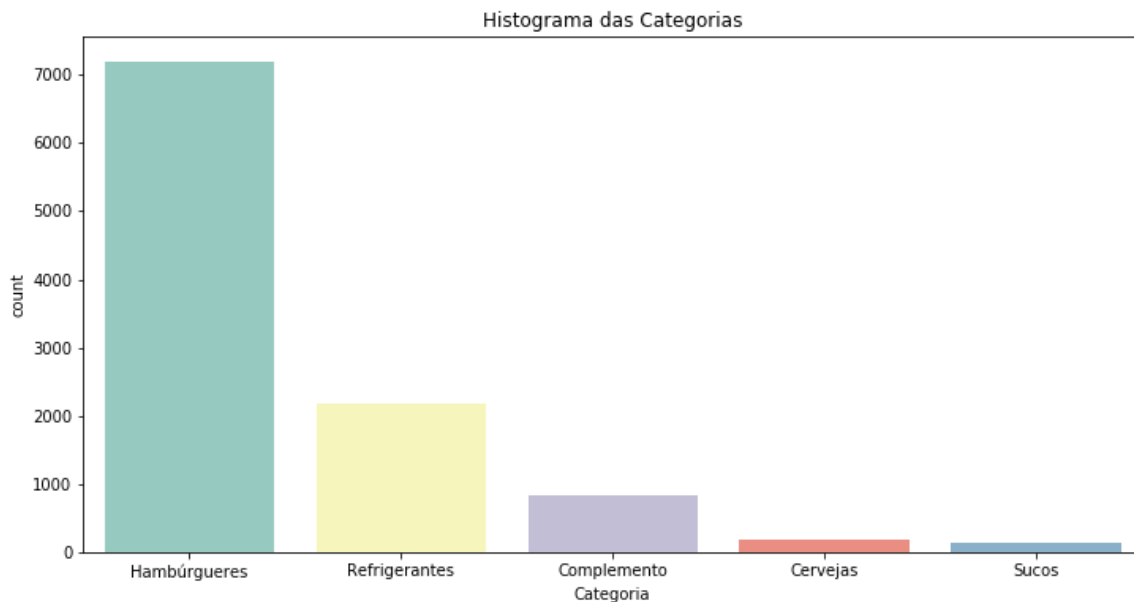
```
Delivery      81.77
Mesa/Comanda  22.67
Balcão        0.80
Caixa         0.08
Name: TipoPedido, dtype: float64
```

In [41]:

```
plt.figure(figsize=(12,6))
sns.countplot(x="Categoria", data=df, palette="Set3")
plt.title("Histograma das Categorias")
```

Out[41]:

Text(0.5, 1.0, 'Histograma das Categorias')

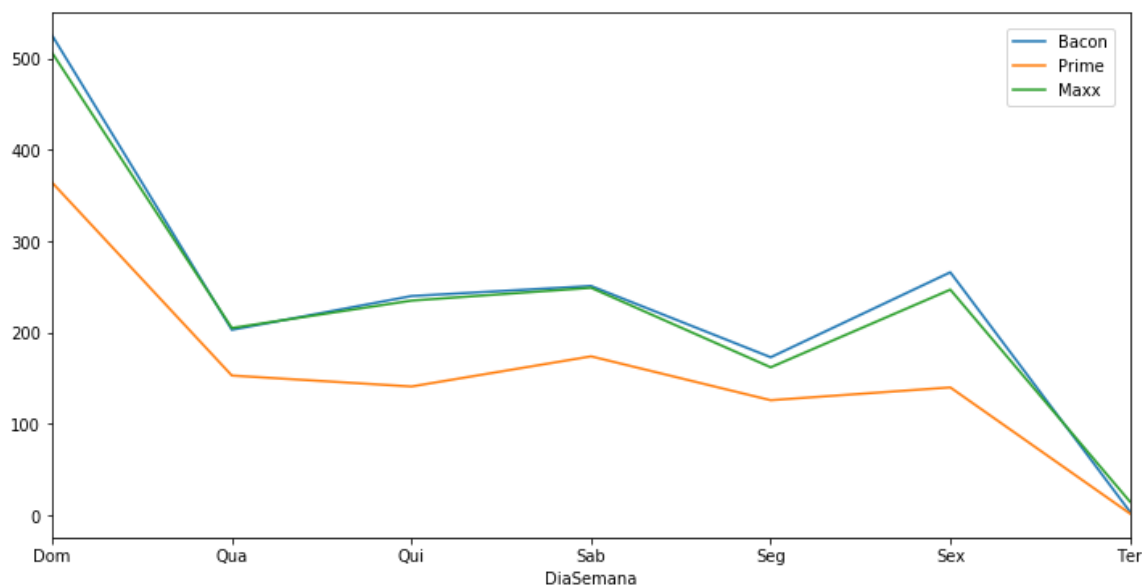


In [42]:

```
plt.figure(figsize=(12,6))
df[df['Produto']=='207 +BACON'].groupby('DiaSemana').count()['Categoria'].plot()
df[df['Produto']=='207 PRIME'].groupby('DiaSemana').count()['Categoria'].plot()
df[df['Produto']=='207 MAXX'].groupby('DiaSemana').count()['Categoria'].plot()
plt.legend(["Bacon", "Prime", "Maxx"], loc = 0, borderaxespad=1)
```

Out[42]:

<matplotlib.legend.Legend at 0xcba6990>

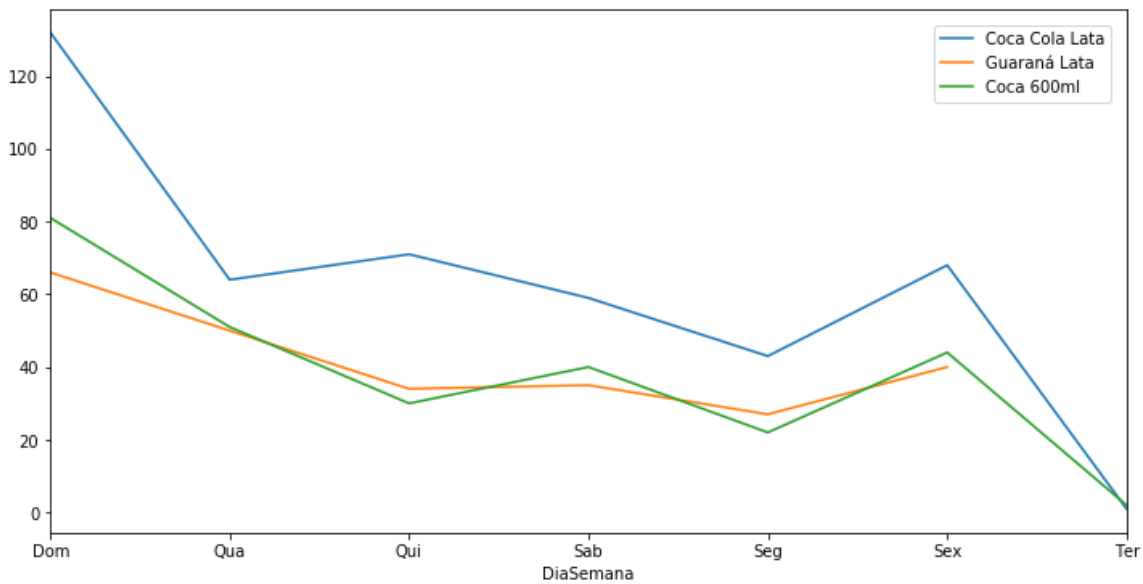


In [43]:

```
plt.figure(figsize=(12,6))
df[df['Produto']=='COCA COLA LATA'].groupby('DiaSemana').count()['Categoria'].plot()
df[df['Produto']=="GUARANÁ ANTÁRTICA LATA"].groupby("DiaSemana").count()["Categoria"].p
lot()
df[df['Produto']=="COCA COLA 600ML"].groupby("DiaSemana").count()["Categoria"].plot()
#df[df['Produto']==""].groupby("DiaSemana").count()["Categoria"].plot()
plt.legend(["Coca Cola Lata", "Guaraná Lata", "Coca 600ml"], loc = 0, borderaxespad=1)
```

Out[43]:

<matplotlib.legend.Legend at 0xcd91030>

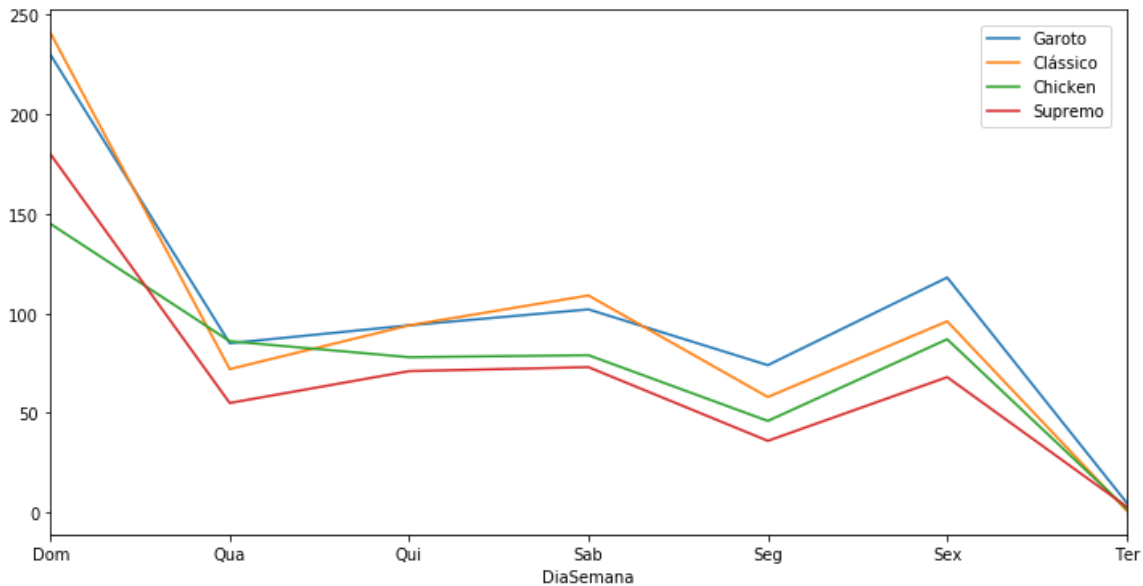


In [44]:

```
plt.figure(figsize=(12,6))
df[df['Produto']=="207 GAROTO"].groupby("DiaSemana").count()["Categoria"].plot()
df[df['Produto']=="207 CLÁSSICO"].groupby("DiaSemana").count()["Categoria"].plot()
df[df['Produto']=="207 CHICKEN"].groupby("DiaSemana").count()["Categoria"].plot()
df[df['Produto']=="207 SUPREMO"].groupby("DiaSemana").count()["Categoria"].plot()
plt.legend(["Garoto", "Clássico", "Chicken", "Supremo"], loc=0, borderaxespad=1)
```

Out[44]:

<matplotlib.legend.Legend at 0xcb1e990>



In []: