Exercícios - Séries Temporais

Questão 1

Utilizando o *dataset* `covid_saopaulo.csv`, plote o gráfico do número de casos de COVID em SP e a média móvel dos últimos 15 dias.

Resolução:

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sp = pd.read_csv('covid_saopaulo.csv')
sp.sort_values(by='date', inplace=True)
sp.head(3)
```

Out[1]:

	date	confirmed	deaths
0	2020-02-26	0	0
1	2020-02-27	0	0
2	2020-02-28	1	0

In [2]:
```
sp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 302 entries, 0 to 301
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   date       302 non-null    object
 1   confirmed  302 non-null    int64 
 2   deaths     302 non-null    int64 
dtypes: int64(2), object(1)
memory usage: 9.4+ KB
```

In [3]:
```
sp['date'] = pd.to_datetime(sp['date'])
sp.info()
```

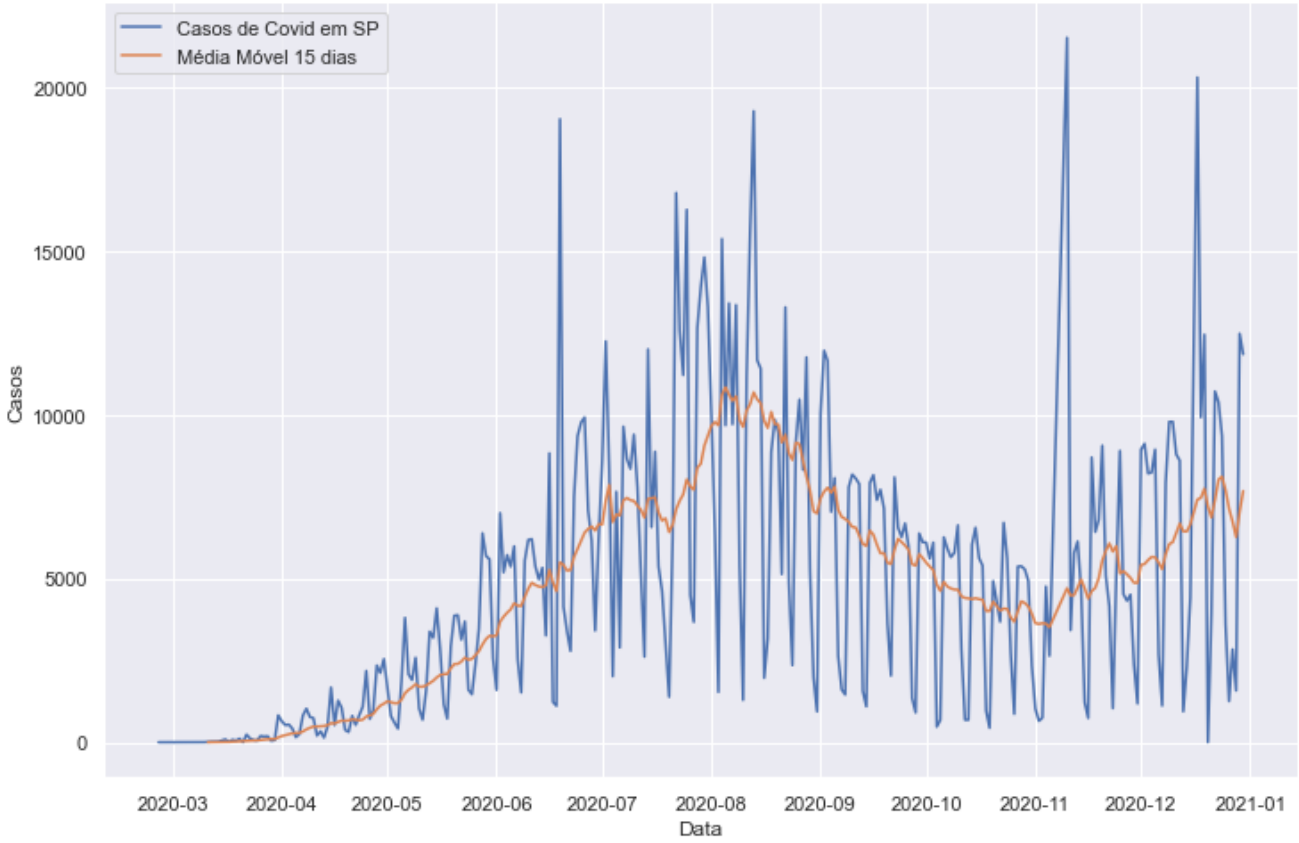
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 302 entries, 0 to 301
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype         
---  ------     --------------  -----         
 0   date       302 non-null    datetime64[ns]
 1   confirmed  302 non-null    int64         
 2   deaths     302 non-null    int64         
dtypes: datetime64[ns](1), int64(2)
memory usage: 9.4 KB
```

In [4]:
```
sp['mm15d'] = sp['confirmed'].rolling(15).mean()
sp.tail(5)
```

Out[4]:

	date	confirmed	deaths	mm15d
297	2020-12-26	1253	13	7131.266667
298	2020-12-27	2836	55	6734.266667
299	2020-12-28	1576	39	6265.933333
300	2020-12-29	12477	293	7035.066667
301	2020-12-30	11849	282	7670.800000

In [5]:
```
sns.set_theme()
sns.set()
plt.figure(figsize=(12,8))
fig1 = sns.lineplot(data=sp, x='date', y='confirmed', label='Casos de Covid em SP')
fig2 = sns.lineplot(data=sp, x='date', y='mm15d', label='Média Móvel 15 dias')
fig1.set(xlabel='Data',ylabel='Casos')
fig2.set(xlabel='Data',ylabel='Casos')
plt.show()
```



Questão 2

Ainda utilizando o *dataset* `covid_saopaulo.csv`, agora para os casos de óbitos por COVID em SP, define as componentes da série temporal (sazonalidade, tendência, resíduo).

Resolução:

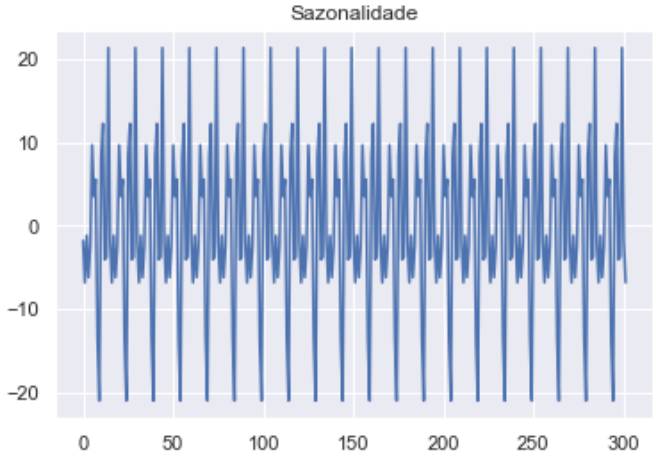
In [6]:
```
from statsmodels.tsa.seasonal import seasonal_decompose

mortes_sp = sp.drop(['confirmed', 'mm15d'], axis=1)
mortes_sp.head(3)
```

Out[6]:

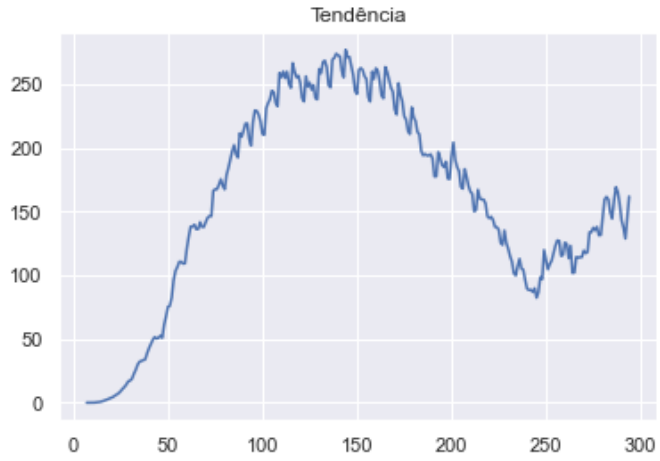
	date	deaths
0	2020-02-26	0
1	2020-02-27	0
2	2020-02-28	0

In [7]:
```
comps_st = seasonal_decompose(mortes_sp['deaths'], period=15)
print('Sazonalidade:')
sazon = comps_st.seasonal
display(sazon.head(3))
sazon.plot(title='Sazonalidade')
plt.show()

tend = comps_st.trend.dropna()
print('Tendência:')
display(tend.head(3))
tend.plot(title='Tendência')
plt.show()

residuo = comps_st.resid.dropna()
print('Resíduo:')
display(residuo.head(3))
residuo.plot(title='Resíduo')
plt.show()
```

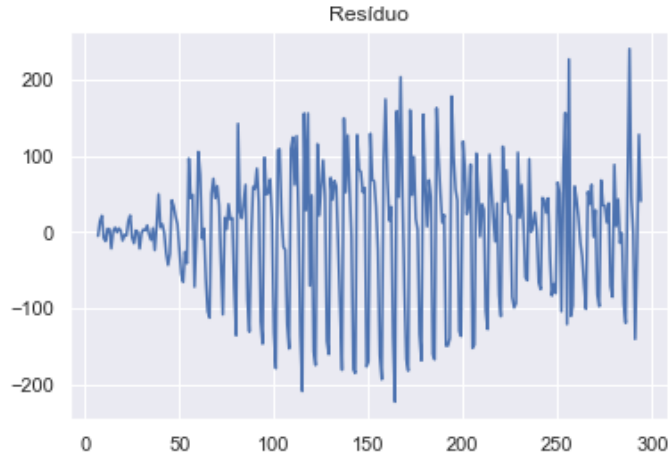
```
Sazonalidade:
0   -1.792503
1   -6.806538
2   -1.178468
Name: seasonal, dtype: float64
```




```
Tendência:
7    0.0
8    0.0
9    0.0
Name: trend, dtype: float64
```




```
Resíduo:
7    -5.567497
8    13.609170
9    20.999170
Name: resid, dtype: float64
```



Questão 3

Plote o gráfico da série temporal e defina as componentes para o *dataset* `airline_passengers.csv`, dado que para o caso deste *dataset* temos como período mensal e sazonalidade multiplicativa.

Resolução:

In [8]:
```
airpass = pd.read_csv('airline-passengers.csv', delimiter=',')
display(airpass.head(3))
```

```
airpass.describe()
airpass.info()
```

	Month	International airline passengers: monthly totals in thousands. Jan 49 ? Dec 60
0	1949-01	112
1	1949-02	118
2	1949-03	132

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 144 entries, 0 to 143
Data columns (total 2 columns):
 #   Column                                                                  Non-Null Count  Dtype 
---  ------                                                                  --------------  ----- 
 0   Month                                                                   144 non-null    object
 1   International airline passengers: monthly totals in thousands. Jan 49 ? Dec 60  144 non-null    int64 
dtypes: int64(1), object(1)
memory usage: 2.4+ KB
```

In [9]:
```
airpass.rename(columns={'Month':'Mes','International airline passengers: monthly totals in thousands. Jan 49 ? Dec 60':'Qty'},inplace=True)
airpass.head(3)
```

Out[9]:

	Mes	Qty
0	1949-01	112
1	1949-02	118
2	1949-03	132

In [10]:
```
airpass.set_index('Mes', inplace=True)
airpass.head(3)
```

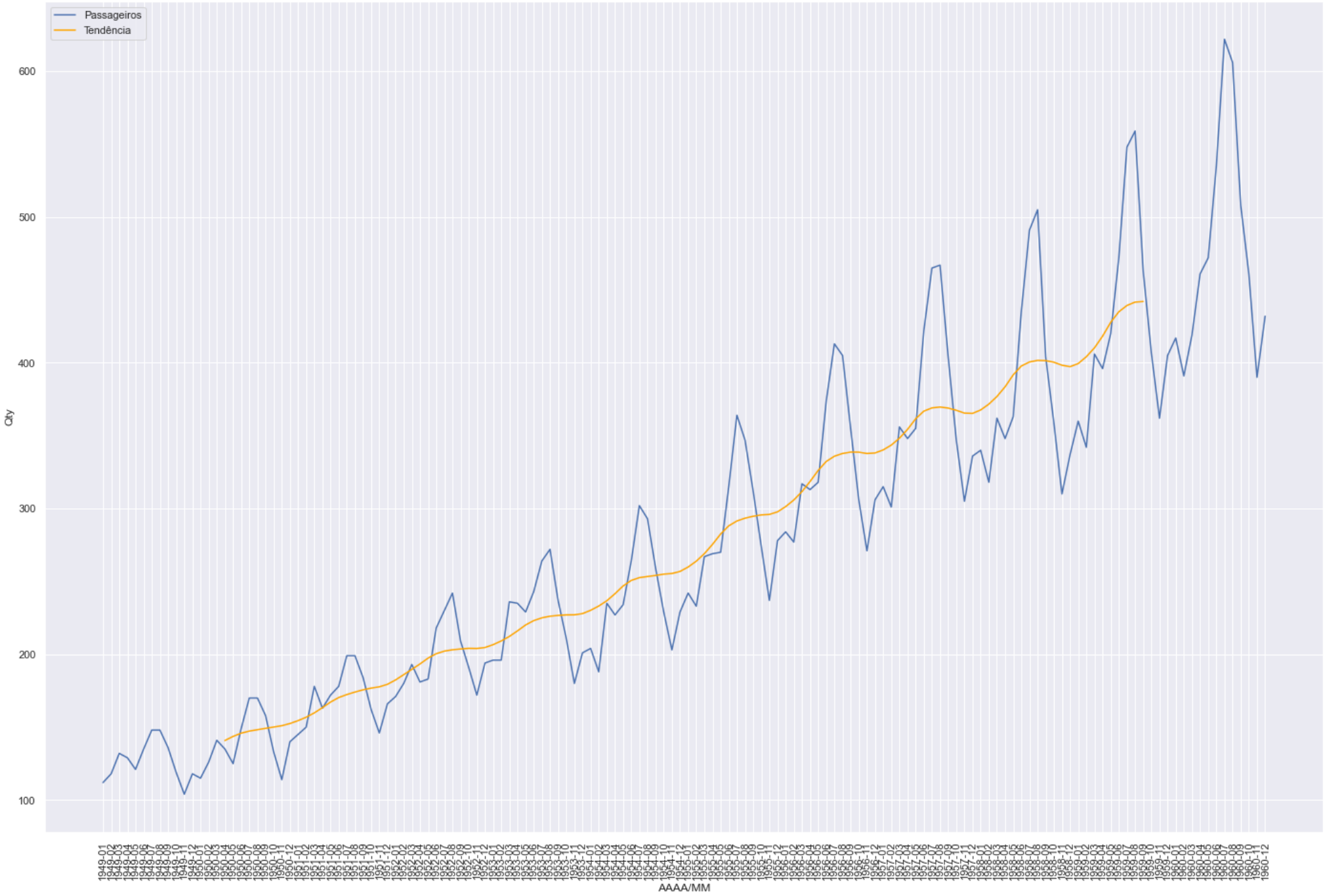
Out[10]:

	Qty
Mes	
1949-01	112
1949-02	118
1949-03	132

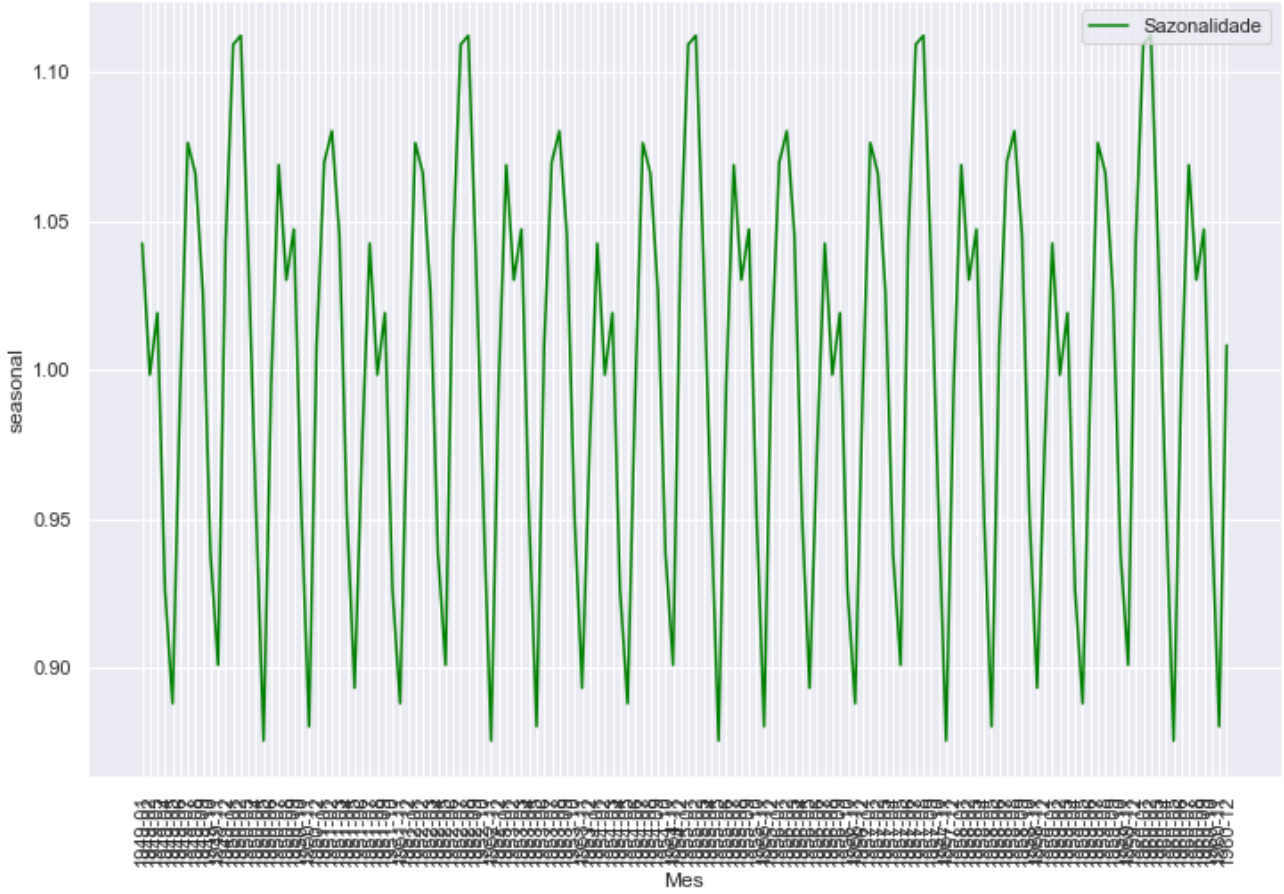
In [30]:
```
comps_airpass = seasonal_decompose(airpass, period=30, model='multiplicative')
#comps_airpass.plot();
```

In [34]:
```
plt.figure(figsize=(24,16))
fig1 = sns.lineplot(data=airpass, x='Mes', y='Qty', label='Passageiros')
sns.lineplot(data=comps_airpass.trend, label='Tendência', color='orange')
fig1.set(xlabel='AAAA/MM',ylabel='Qty')
```

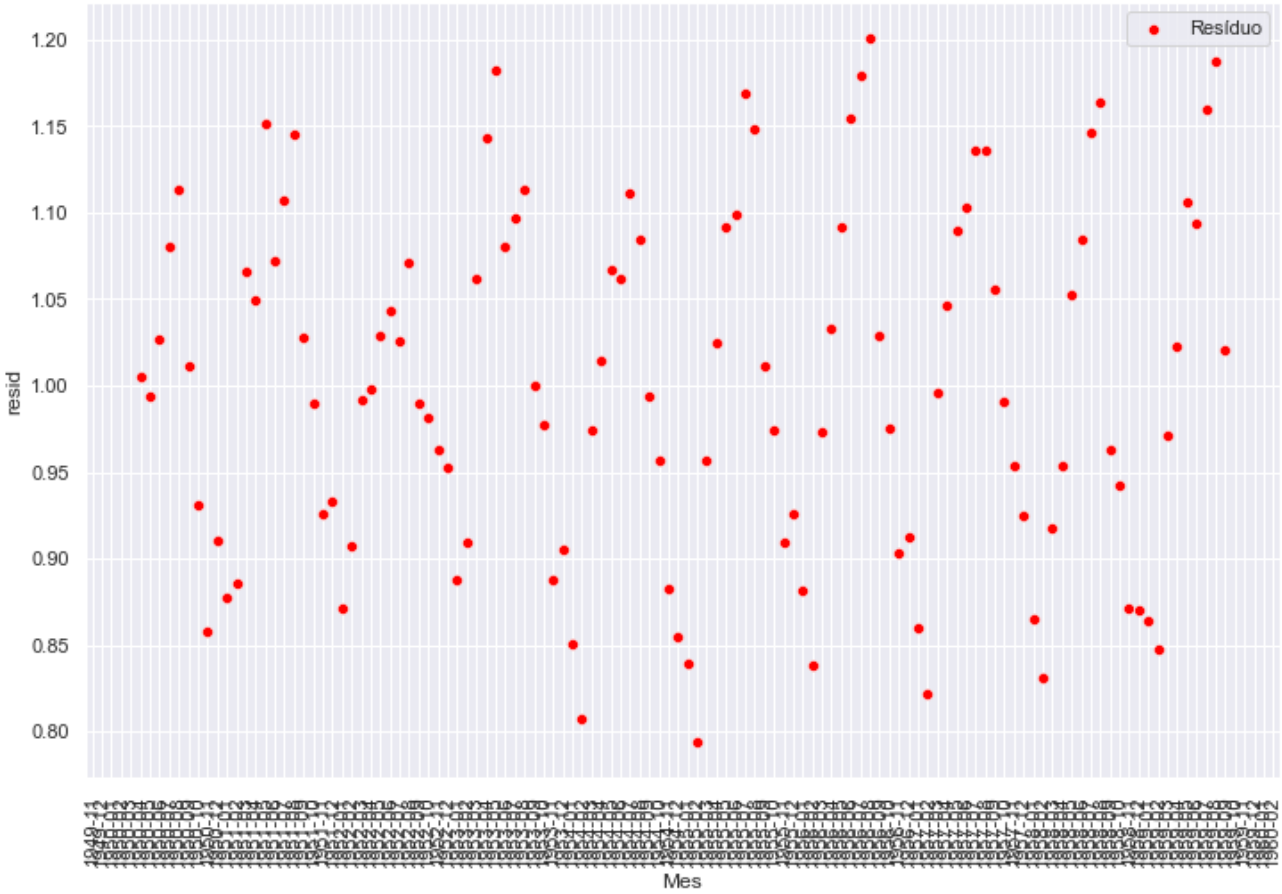
```
fig1.tick_params(axis='x', rotation=90)
plt.show()
```



In [36]:
```
plt.figure(figsize=(12,8))
fig1 = sns.lineplot(data=comps_airpass.seasonal, label='Sazonalidade', color='green')
fig1.tick_params(axis='x', rotation=90)
plt.show()
```



In [37]:
```
plt.figure(figsize=(12,8))
fig1 = sns.scatterplot(data=comps_airpass.resid, label='Resíduo', color='red')
fig1.tick_params(axis='x', rotation=90)
plt.show()
```


Resíduo
1.20
1.15
1.10
1.05
1.00
0.95
0.90
0.85
0.80
resid
Mes