In [1]:
```
import datacleaner
import config
import os
import sys
import pandas as pd
import numpy as np
from fbprophet import Prophet
import matplotlib.pyplot as plt
import fbprophet

# "high resolution"
%config InlineBackend.figure_form
```

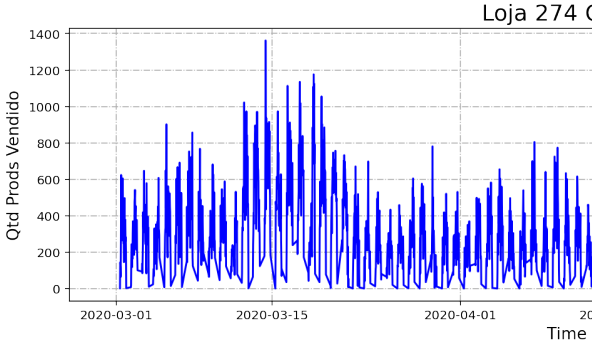
Análise dados para loj

Leitura do ficheiro para cri as variáveis número de ver (sales, clients)

In [2]:
```
df_274_time_sale = datacleaner.ge
df_274_time_cli = datacleaner.get
df_274_time_sale.reset_index(inpl
df_274_time_cli.reset_index(inpla
```

Visualizando os dados reais para vendas

In [3]:
```
plt.figure(figsize=(16,4))
plt.grid(linestyle='-.')
plt.plot(df_274_time_sale.ds, df_
plt.title('Loja 274 Qtd Produtos/
plt.ylabel('Qtd Prods Vendido', f
plt.xlabel('Time (aaaa-mm-dd)', f
plt.show()
```



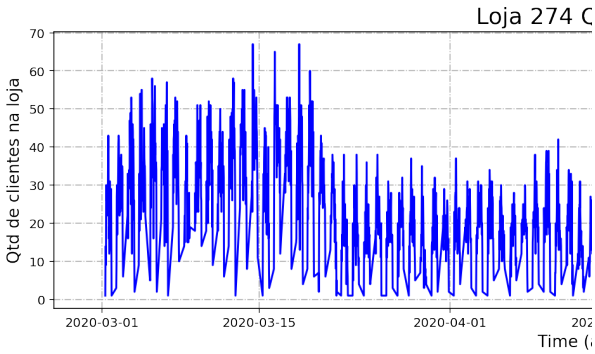
In [4]:
```
# Obtendo o valor máximo de produ

df_274_time_sale['y'].max()
```

Out[4]: 1362

Visualizando os dados reais para clientes

In [5]:
```
plt.figure(figsize=(16,4))
plt.grid(linestyle='-.')
plt.plot(df_274_time_cli.ds, df_2
plt.title('Loja 274 Qtd Clientes/
plt.ylabel('Qtd de clientes na lo
plt.xlabel('Time (aaaa-mm-dd)', f
plt.show()
```



In [6]:
```
# Obtendo o número máximo de clie
df_274_time_cli['y'].max()
```

Out[6]: 67

In [7]:
```
# definindo o cap (carrying capac
df_274_time_sale['cap'] = 1362
df_274_time_sale.reset_index(drop
df_274_time_sale
```

Out[7]:

	ds	y	cap
0	2020-03-01 07:30:00	2	1362
1	2020-03-01 08:00:00	144	1362
2	2020-03-01 08:30:00	117	1362
3	2020-03-01 09:00:00	65	1362
4	2020-03-01 09:30:00	271	1362
...	...	...	...
2550	2020-05-31 18:30:00	268	1362
2551	2020-05-31 19:00:00	173	1362
2552	2020-05-31 19:30:00	136	1362
2553	2020-05-31 20:00:00	23	1362
2554	2020-05-31 20:30:00	80	1362

2555 rows × 3 columns

In [8]:
```
# df_274_time_sale['floor'] = ??
df_274_time_cli['cap'] = 67
df_274_time_cli.reset_index(drop=
# df_274_time_cli['floor'] = ??
```

Out[8]:

	ds	y	cap
0	2020-03-01 07:30:00	1	67
1	2020-03-01 08:00:00	11	67
2	2020-03-01 08:30:00	14	67
3	2020-03-01 09:00:00	9	67
4	2020-03-01 09:30:00	30	67
...	...	...	...
2550	2020-05-31 18:30:00	18	67
2551	2020-05-31 19:00:00	9	67
2552	2020-05-31 19:30:00	9	67
2553	2020-05-31 20:00:00	3	67
2554	2020-05-31 20:30:00	7	67

2555 rows × 3 columns

Inicializando e ajustan modelos

Paramêtros:

- growth: linear/logistic
- seasonality: additive/multiplicative
- holidays:
- changepoints:
- interval_width: assume que o futuro verá taxas de mudança do passado.

Por padrão o Prophet retornará apena observação.

Para obter a incerteza na sazonalidad bayesiana completa. Isso é feito usan padrão é 0)

**** mcmc = Markov chain Monte Carlo***

In [9]:
```
path = getattr(config,'path','def
old_stdout = sys.stdout
sys.stdout = open(path+'/logs/sis
m_274_time_sale = Prophet(growth=
                          interva
                          changep
                          yearly_
                          weekly_
                          holiday
m_274_time_sale.add_country_holid
m_274_time_sale.fit(df_274_time_s
m_274_time_cli = Prophet(growth='
                         interval
                         changepo
                         yearly_s
                         weekly_s
                         holidays
m_274_time_cli.add_country_holida
m_274_time_cli.fit(df_274_time_cl
```

Out[9]:
```
<fbprophet.forecaster.Prophet at
```

Parâmetro growth (saturação)

- Quando a previsão cresce, alguns ponto
  um teto, isso é chamado de carrying cap
  mercados, cujos limites podem ser atinç
  possibilidades como novas tecnologias,
  econômicas, mudanças de hábitos e etc
  da previsão para que o modelo comport
- É possível definir o carrying capacity (ca
  do dataframe.

Por padrão o Prophet usa um modelo de s
o modelo logistico.

Parâmetro interval_width

- O interval_width de confiança = 95%; iss
  produzir um intervalo de confiança em t

Parâmetros Trend Changepoints

- Para os dados que estamos a analisar, c
  momento ou índice de tempo que define
  mudar sua direção, quer seja crescente
  de inflexão.
- Prophet usa um entre dois métodos para
  - Especificar a flexibilidade das tendê
  - Especificar o local dos changepoint
    próprio dataframe como uma série

Parâmetro changepoint_prior_scale

- Representa o quão flexível o modelo irá

Parâmetros de sazonalidade

- Definem o período de sazonalidade a se
  desejamos que o algoritmo considere sa
  um evento) anual e mensal.

Parâmetro holiday

- Os feriados e eventos influenciam no co
  das pessoas. Para esse estudo foram ac
  conhecidos, entretanto, cabe destacar c
  municipais que impactam o comércio lo

In [10]:
```
m_274_time_sale.train_holiday_nam
```

Out[10]:
```
0                        parana
1                New Year's Day
2                    Tiradentes
3                  Worker's Day
4              Independence Day
5    Our Lady of the Apparition
6                All Souls' Day
7     Republic Proclamation Day
8                      Christmas
dtype: object
```

Criação do dataframe para

- Identificando os valores criados fora do
- Frequência horária
- Cada 30 minutos de cada hora
- Carry Capacity (teto) = 1362 (valor carec
  - Carry Capacity (base) = Não utilizac

In [11]:
```

future_274_time_sale = m_274_time
future_274_time_sale['cap'] = 136
teste_future_274_time_sale = futu
teste_future_274_time_sale
```

Out[11]:

	ds	cap
2558	2020-06-01 00:30:00	1362
2559	2020-06-01 01:30:00	1362
2560	2020-06-01 02:30:00	1362
2561	2020-06-01 03:30:00	1362
2562	2020-06-01 04:30:00	1362
...	...	...
4350	2020-08-14 16:30:00	1362
4351	2020-08-14 17:30:00	1362
4352	2020-08-14 18:30:00	1362
4353	2020-08-14 19:30:00	1362
4354	2020-08-14 20:30:00	1362

1797 rows × 2 columns

O prophet está projetando dados para hor
isto é, após 20:30 até 07:00, como não há
podem impactar nas análises. Dessa form
necessário remover do dataframe, projeta

In [12]:
```
future_274_time_sale_adjusted = f
future_274_time_sale_adjusted['ds
future_274_time_sale_adjusted = f
future_274_time_sale_adjusted = f
#future_274_time_sale_adjusted.re
future_274_time_sale_adjusted
```

Out[12]:

	ds	cap
ds		
2020-03-01 07:30:00	2020-03-01 07:30:00	136
2020-03-01 08:00:00	2020-03-01 08:00:00	136
2020-03-01 08:30:00	2020-03-01 08:30:00	136
2020-03-01 09:00:00	2020-03-01 09:00:00	136
2020-03-01 09:30:00	2020-03-01 09:30:00	136
...	...	..
2020-08-14 16:30:00	2020-08-14 16:30:00	136
2020-08-14 17:30:00	2020-08-14 17:30:00	136
2020-08-14 18:30:00	2020-08-14 18:30:00	136
2020-08-14 19:30:00	2020-08-14 19:30:00	136
2020-08-14 20:30:00	2020-08-14 20:30:00	136

3473 rows × 2 columns

In [13]:
```
# Confirmando que os valores fora
teste2_future_274_time_sale = fut
teste2_future_274_time_sale
```

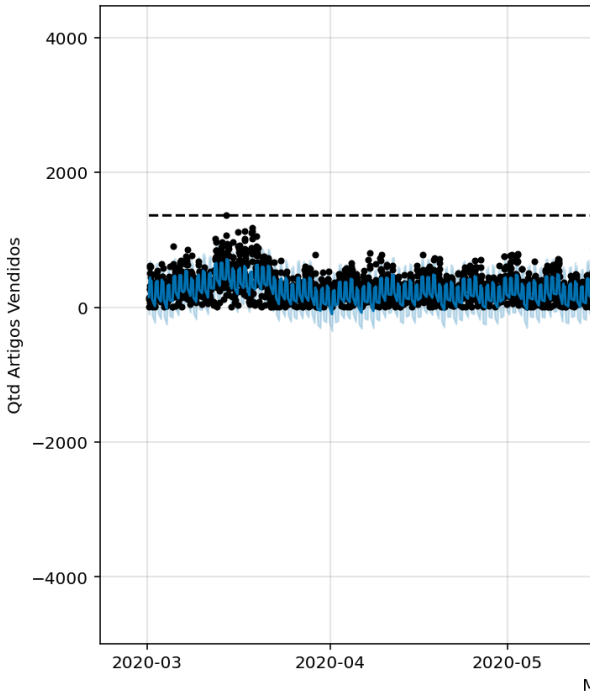
Out[13]:

	ds	cap
ds		
2020-06-01 07:30:00	2020-06-01 07:30:00	136
2020-06-01 08:30:00	2020-06-01 08:30:00	136
2020-06-01 09:30:00	2020-06-01 09:30:00	136
2020-06-01 10:30:00	2020-06-01 10:30:00	136
2020-06-01 11:30:00	2020-06-01 11:30:00	136
...	...	..
2020-08-14 16:30:00	2020-08-14 16:30:00	136
2020-08-14 17:30:00	2020-08-14 17:30:00	136
2020-08-14 18:30:00	2020-08-14 18:30:00	136
2020-08-14 19:30:00	2020-08-14 19:30:00	136
2020-08-14 20:30:00	2020-08-14 20:30:00	136

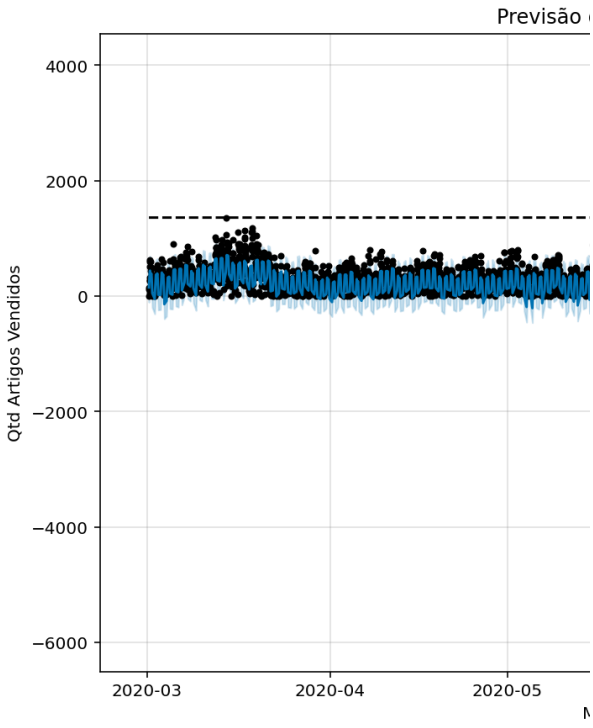
1050 rows × 2 columns

Visualizando o gráfico da e vendas

In [14]:
```
forecast_time_sale = m_274_time_s
fig_time_sale = m_274_time_sale.p
plt.title('Previsão de Vendas Loj
plt.show()
```

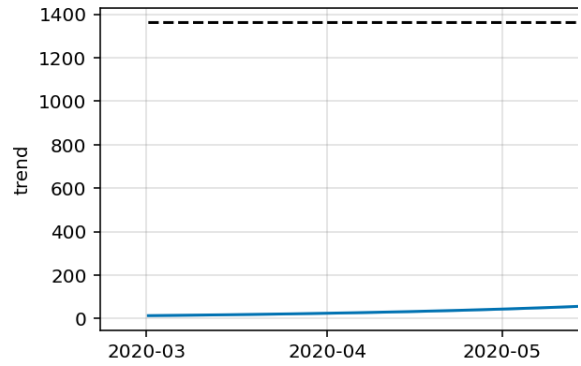


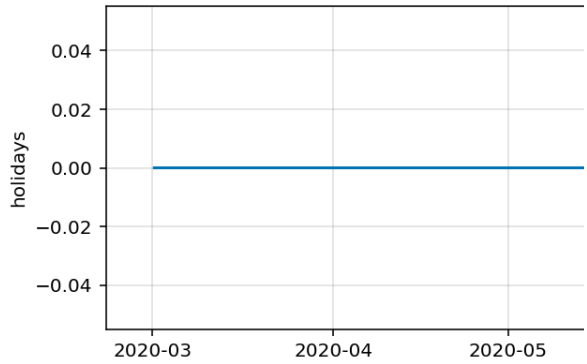
In [15]:
```
# Previsão sem a remoção dos valo
forecast_teste = m_274_time_sale.
fig_time_gap_sale = m_274_time_sa
plt.title('Previsão de Vendas Loj
plt.show()
```

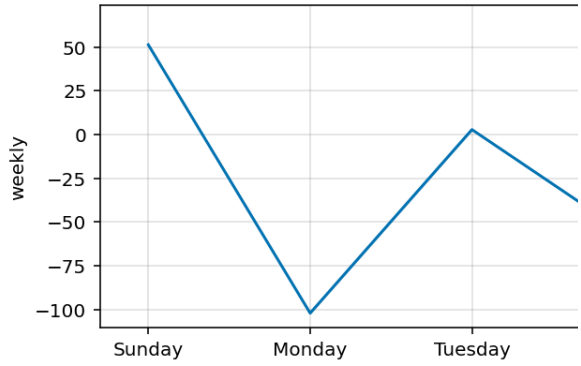


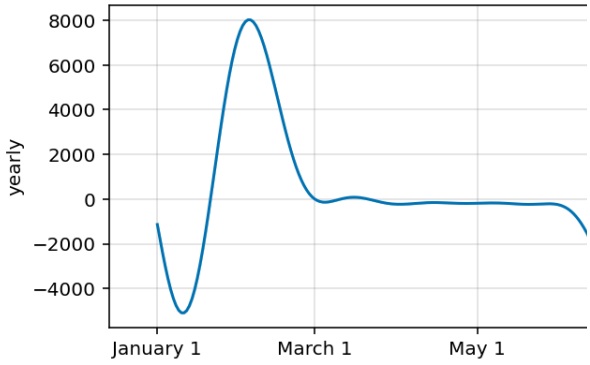
**O dataframe projetado para um futuro cor
minutos) passa a projetar valores mais dis
em função das incertezas que precisam s
uncertainty=True assume uma média da fr
verificada no passado, que seja a mesma
valores negativos estimados no gráfico).**

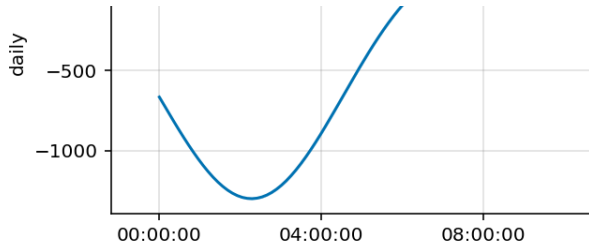
In [16]:
```
fig_comp_time_sale = m_274_time_s
```











Curiosamente o componente diário (daily)
preciso verificar isso

Verificando a quantidade d previstos

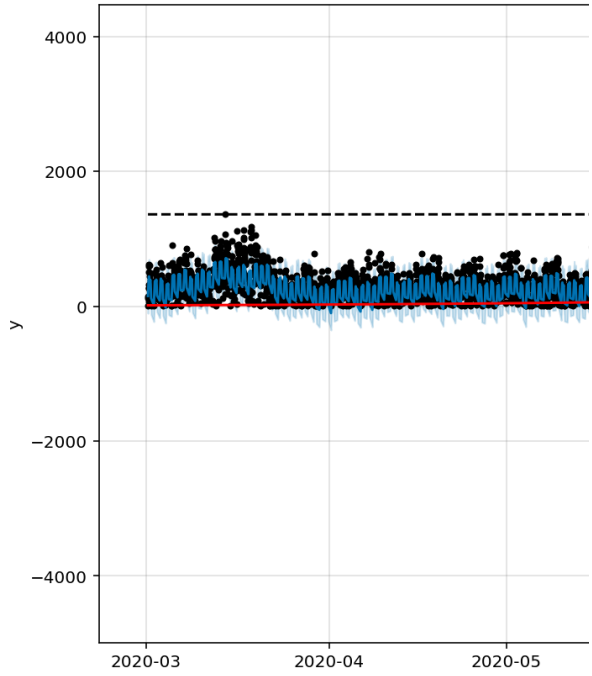
In [17]:
```
#forecast_time_sale = m_274_time_
future_no_art_sales_pred = foreca
future_no_art_sales_pred = future
future_no_art_sales_pred.sample(5
```

Out[17]:

	ds	yhat	yl
3340	2020-08-05 14:30:00	-2047.538233	-230
2505	2020-06-06 19:30:00	-246.577136	-49
3379	2020-08-08 11:30:00	-2236.730794	-250
3090	2020-07-18 16:30:00	3782.366027	353
2571	2020-06-11 15:30:00	-1087.363411	-133

Trend Changepoints

In [18]:
```
from fbprophet.plot import add_ch
fig_274_sales_changepoint = m_274
a = add_changepoints_to_plot(fig_
```



ANÁLISE DOS CLIENT

In [20]:

```
future_274_time_cli = m_274_time_
future_274_time_cli['cap'] = 67
teste_future_274_time_cli = futur
teste_future_274_time_cli
```

Out[20]:

	ds	cap
2558	2020-06-01 00:30:00	67
2559	2020-06-01 01:30:00	67
2560	2020-06-01 02:30:00	67
2561	2020-06-01 03:30:00	67
2562	2020-06-01 04:30:00	67
...	...	...
4350	2020-08-14 16:30:00	67
4351	2020-08-14 17:30:00	67
4352	2020-08-14 18:30:00	67
4353	2020-08-14 19:30:00	67
4354	2020-08-14 20:30:00	67

1797 rows × 2 columns

O prophet está projetando dados para hor
isto é, após 20:30 até 07:00, como não há
podem impactar nas análises. Dessa form
necessário remover do dataframe, projeta

In [21]:
```
future_274_time_cli_adjusted = fu
future_274_time_cli_adjusted['ds'
future_274_time_cli_adjusted = fu
future_274_time_cli_adjusted = fu
#future_274_time_cli_adjusted.res
future_274_time_cli_adjusted
```

Out[21]:

	ds	cap
ds		
2020-03-01 07:30:00	2020-03-01 07:30:00	67
2020-03-01 08:00:00	2020-03-01 08:00:00	67
2020-03-01 08:30:00	2020-03-01 08:30:00	67
2020-03-01 09:00:00	2020-03-01 09:00:00	67
2020-03-01 09:30:00	2020-03-01 09:30:00	67
...	...	...
2020-08-14 16:30:00	2020-08-14 16:30:00	67
2020-08-14 17:30:00	2020-08-14 17:30:00	67
2020-08-14 18:30:00	2020-08-14 18:30:00	67
2020-08-14 19:30:00	2020-08-14 19:30:00	67
2020-08-14 20:30:00	2020-08-14 20:30:00	67

3473 rows × 2 columns

In [22]:
```
# Confirmando que os valores fora
teste2_future_274_time_cli = futu
teste2_future_274_time_cli
```

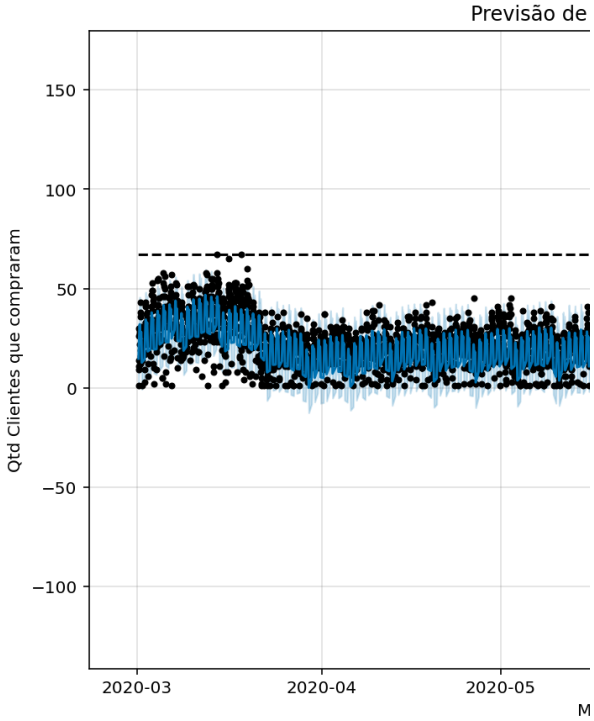
Out[22]:

	ds	cap
ds		
2020-06-01 07:30:00	2020-06-01 07:30:00	67
2020-06-01 08:30:00	2020-06-01 08:30:00	67
2020-06-01 09:30:00	2020-06-01 09:30:00	67
2020-06-01 10:30:00	2020-06-01 10:30:00	67
2020-06-01 11:30:00	2020-06-01 11:30:00	67
...	...	...
2020-08-14 16:30:00	2020-08-14 16:30:00	67
2020-08-14 17:30:00	2020-08-14 17:30:00	67
2020-08-14 18:30:00	2020-08-14 18:30:00	67
2020-08-14 19:30:00	2020-08-14 19:30:00	67
2020-08-14 20:30:00	2020-08-14 20:30:00	67

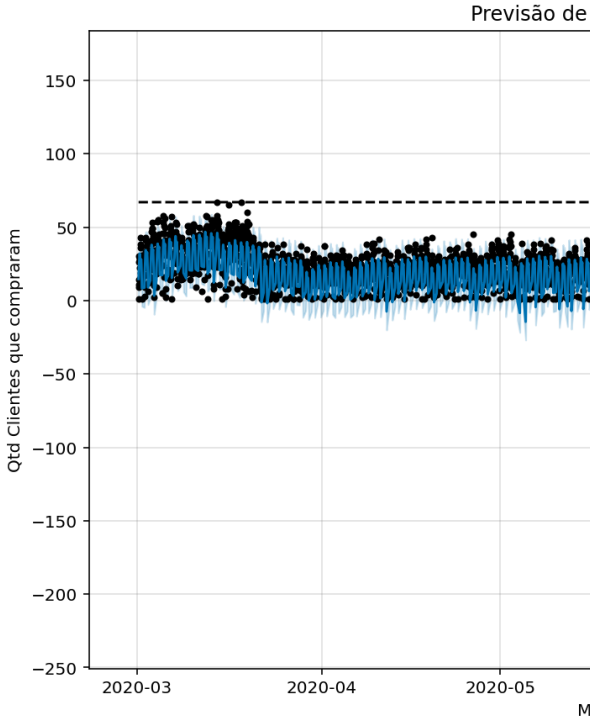
1050 rows × 2 columns

Visualizando o gráfico da e vendas

In [23]:
```
forecast_time_cli = m_274_time_cl
fig_time_cli = m_274_time_cli.plo
plt.title('Previsão de Clientes n
plt.show()
```

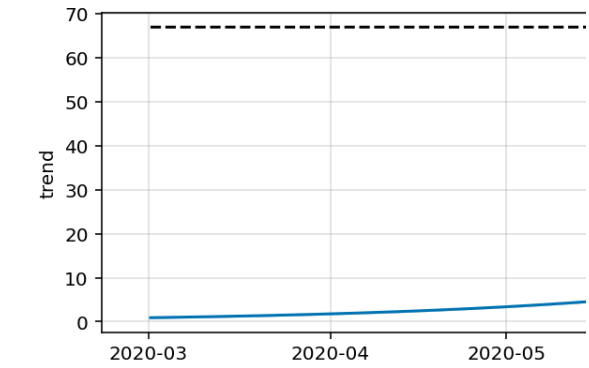


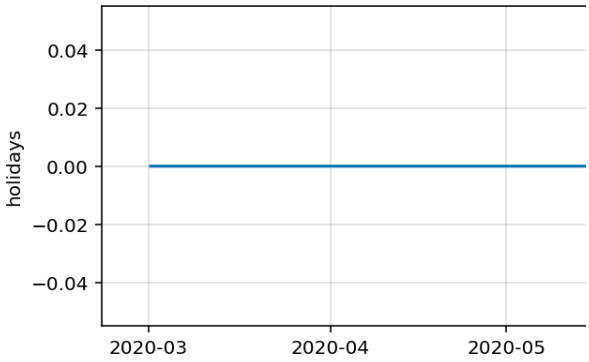
In [24]:
```
# Previsão sem a remoção dos valo
forecast_teste_cli = m_274_time_c
fig_time_gap_cli = m_274_time_cli
plt.title('Previsão de Clientes n
plt.show()
```

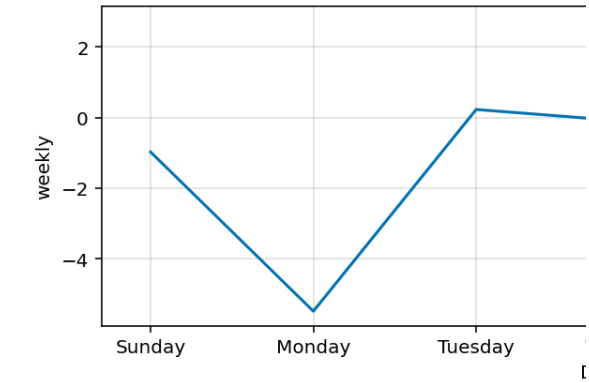


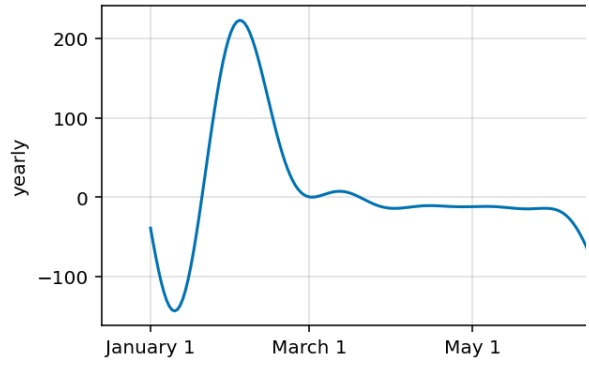
**O dataframe projetado para um futuro cor
minutos) passa a projetar valores mais dis
em função das incertezas que precisam s
uncertainty=True assume uma média da fr
verificada no passado, que seja a mesma
valores negativos estimados no gráfico).**

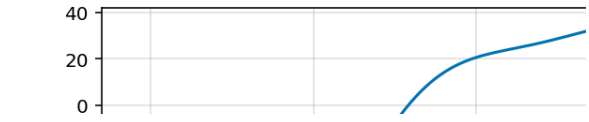
In [25]:
```
fig_comp_time_cli = m_274_time_cl
```

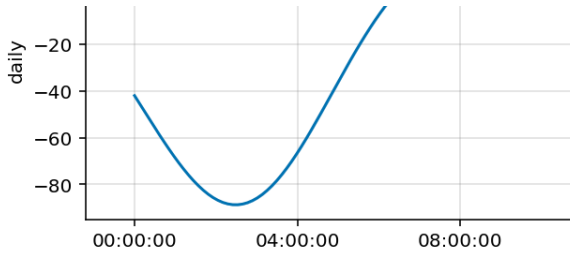












Curiosamente o componente diário (daily)
preciso verificar isso

Verificando a quantidade d
previstos de comprarem na

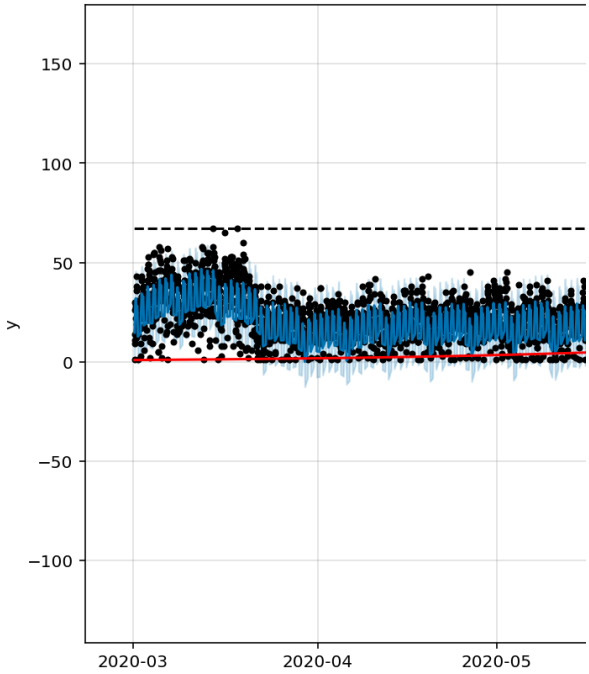
In [26]:
```
#forecast_time_cli = m_274_time_c
future_no_cli_sales_pred = foreca
future_no_cli_sales_pred = future
future_no_cli_sales_pred.sample(5
```

Out[26]:

	ds	yhat	yhat_l
2888	2020-07-04 10:30:00	-28.607160	-41.03
2473	2020-06-04 15:30:00	16.336670	3.34
2969	2020-07-10 07:30:00	48.615955	36.09
2834	2020-06-30 12:30:00	-71.806913	-84.09
3004	2020-07-12 14:30:00	88.972830	74.53

Trend Changepoints

In [27]:
```
from fbprophet.plot import add_ch
fig_274_cli_changepoint = m_274_t
a = add_changepoints_to_plot(fig_
```

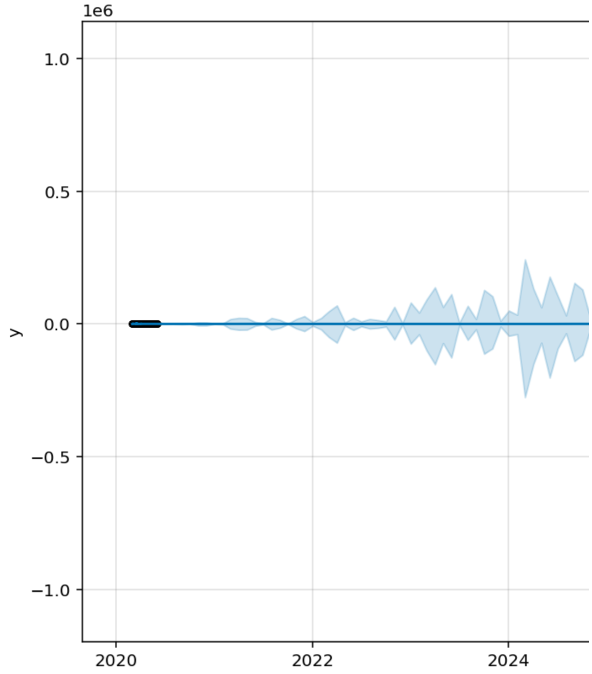


Sales monthly

In [28]:
```
# growth=linear
m_month_time_sale_additive = Prop
                             chang
                             yearl
                             weekl
                             holid
m_month_time_sale = Prophet(inter
                             chang
                             yearl
                             weekl
                             seaso
                             holid
```

In [29]:
```
future_month_time_sale = m_month_
```

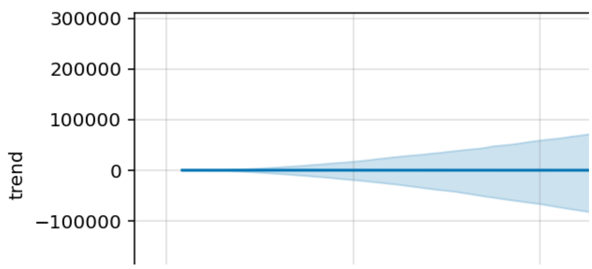
In [30]:
```
fcast_m = m_month_time_sale.predi
fig_fcast_m = m_month_time_sale.p
```

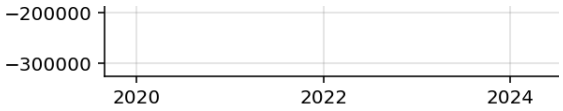


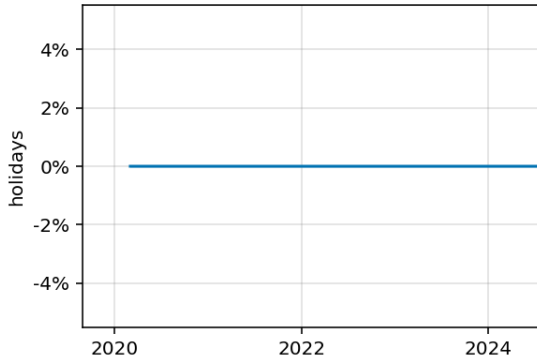
Visualizando os componentes do gráfico

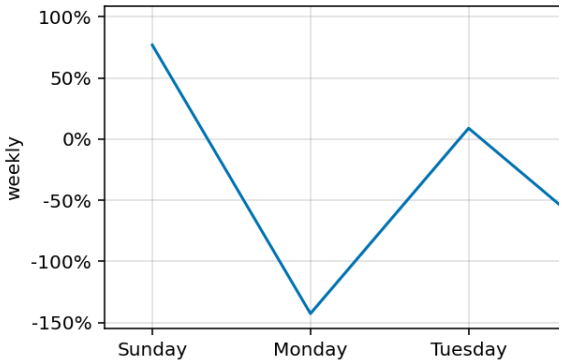
In [31]:
```
fig_fcast_m_comp = m_month_time_s
```

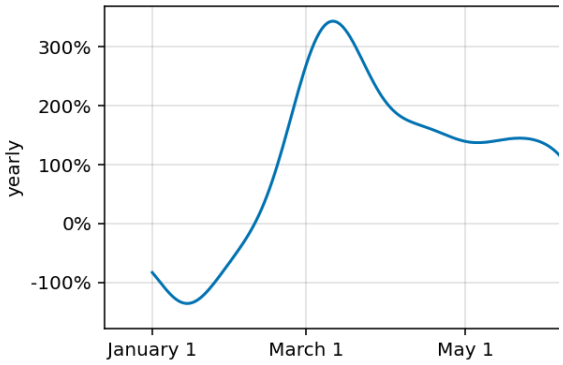
```
/Users/clonyjr/Library/Mobile Doc
/UA/CLONY/Bolsas/Forecast/prophet
ophet/plot.py:413: UserWarning: F
together with FixedLocator
  ax.set_yticklabels(yticklabels)
/Users/clonyjr/Library/Mobile Doc
/UA/CLONY/Bolsas/Forecast/prophet
ophet/plot.py:413: UserWarning: F
together with FixedLocator
  ax.set_yticklabels(yticklabels)
/Users/clonyjr/Library/Mobile Doc
/UA/CLONY/Bolsas/Forecast/prophet
ophet/plot.py:413: UserWarning: F
together with FixedLocator
  ax.set_yticklabels(yticklabels)
/Users/clonyjr/Library/Mobile Doc
/UA/CLONY/Bolsas/Forecast/prophet
ophet/plot.py:413: UserWarning: F
together with FixedLocator
  ax.set_yticklabels(yticklabels)
```

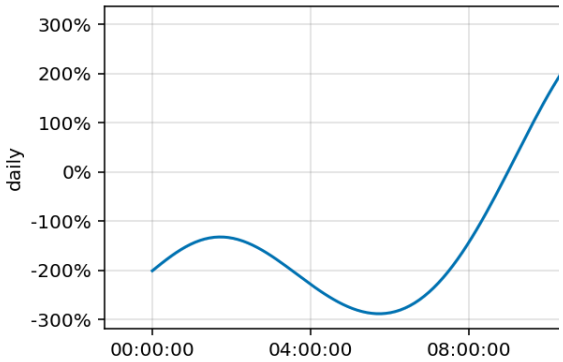












Estimando a quatidade de

In [32]:
```
future_274_time_cli = m_274_time_
future_274_time_cli.tail()
```

Out[32]:

	ds
2915	2021-05-27 20:30:00
2916	2021-05-28 20:30:00
2917	2021-05-29 20:30:00
2918	2021-05-30 20:30:00
2919	2021-05-31 20:30:00

TESTE PARA A DEFINI CHANGEPOINTS (AIND

In [33]:
```
mean = df_274_time_sale['y'].mean
stdev = df_274_time_sale['y'].std
quantile1 = df_274_time_sale['y']
quantile2 = df_274_time_sale['y']
iqr = quantile2 - quantile1
high = mean + stdev
low = mean - stdev
```

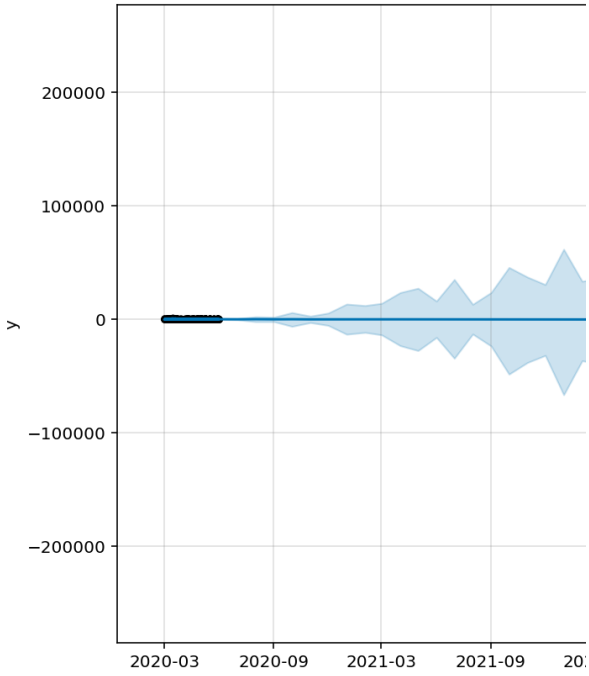
In [34]:
```
# Essa é a definição dos changepo
# desvio padrão
df_time_sale_filtered = df_274_ti
df_time_sale_filtered_changepoint

# Essa é a definição dos changepo
filtered_iqr = df_274_time_sale[(
```

TESTE PARA OS PARÂ SAZONALIDADE

In [35]:
```
m_274_sale_test = Prophet(interva
m_274_sale_test.fit(df_274_time_s
future_274_sales_test = m_274_sal
forecast_274_sales_test = m_274_s
fig_274_sales_test = m_274_sale_t
```

```
INFO:fbprophet:Disabling yearly s
ly_seasonality=True to override t
```



In [17]:
```
m_274_sale_test = Prophet(seasona
m_274_sale_test.fit(df_274_time_s
future_274_sales_test = m_274_sal
forecast_274_sales_test = m_274_s
fig_274_sales_test = m_274_sale_t
```

```
INFO:fbprophet:Disabling yearly s
ly_seasonality=True to override t
```