Final_MainP2

June 21, 2020

[1]:
```
# Importamos paquetes y cosas importantes:

from ORM_database import Quarantine, Comuna, TimeSerie
import matplotlib.pyplot as plt
import pandas as pd
```

Definimos algunas funciones que realizan las consultas y serán útiles más adelante:

[2]:
```
def get_contagiados(id_comuna):
    '''
    Obtiene un dataframe con los contagiados totales por  comuna,
    consultando la base de datos previamente construida
    :param id_comuna: entero con el id de la comuna
    :return: dataframe
    '''
    query = (TimeSerie.
             select(TimeSerie.date, TimeSerie.value).
             where((TimeSerie.comuna_id == id_comuna) & (TimeSerie.serie_id ==␣
 ↪0)))
    df = pd.DataFrame(list(query.dicts()))
    df['date'] = pd.to_datetime(df['date'])
    df = df.set_index('date')
    return df


def get_cuarentenas(id_comuna):
    """
    Obtiene un dataframe con las cuarentenas aplicadas a cada comuna,
    consultando la base de datos previamente construida
    :param id_comuna: entero con el id de la comuna
    :return: dataframe
    """
    query = (Quarantine.
             select(Quarantine.init_day, Quarantine.end_day).
             where((Quarantine.comuna_id == id_comuna)))
    df = pd.DataFrame(list(query.dicts()))
    df['init_day'] = pd.to_datetime(df['init_day'])
    df['end_day'] = pd.to_datetime(df['end_day'])
```

```
    return df


def get_movilidad(id_comuna):
    """
    Obtiene un dataframe con los índices de movilidad de cada comuna,
    consultando la base de datos previamente construida
    :param id_comuna: entero con el id de la comuna
    :return: dataframe
    """
    query = (TimeSerie.
             select(TimeSerie.date, TimeSerie.value).
             where((TimeSerie.comuna_id == id_comuna) & (TimeSerie.serie_id ==␣
 ↪1)))
    df = pd.DataFrame(list(query.dicts()))
    df['date'] = pd.to_datetime(df['date'])
    df = df.set_index('date')
    return df

def get_sintomas(id_comuna):
    """
    Obtiene un dataframe con casos de contagiados nuevos por fecha de inicio de␣
 ↪síntomas por comuna, por semana epidemiológica
    consultando la base de datos previamente construida
    :param id_comuna: entero con el id de la comuna
    :return: dataframe
    """
    query = (TimeSerie.
             select(TimeSerie.date, TimeSerie.value).
             where((TimeSerie.comuna_id == id_comuna) & (TimeSerie.serie_id ==␣
 ↪2)))
    df = pd.DataFrame(list(query.dicts()))
    df['date'] = pd.to_datetime(df['date'])
    df = df.set_index('date')
    return df


def get_all(id_comuna):
    """
    Ejecuta todas las funciones anteriores
    :param id_comuna: entero con el id de cada comuna
    :return: cuatro dataframes con la información pedida
    """
    c = get_contagiados(id_comuna)
    q = get_cuarentenas(id_comuna)
    m = get_movilidad(id_comuna)
    s = get_sintomas(id_comuna)
```

```
    return c, q, m, s
```

Definimos las funciones para realizar los gráficos de interés:

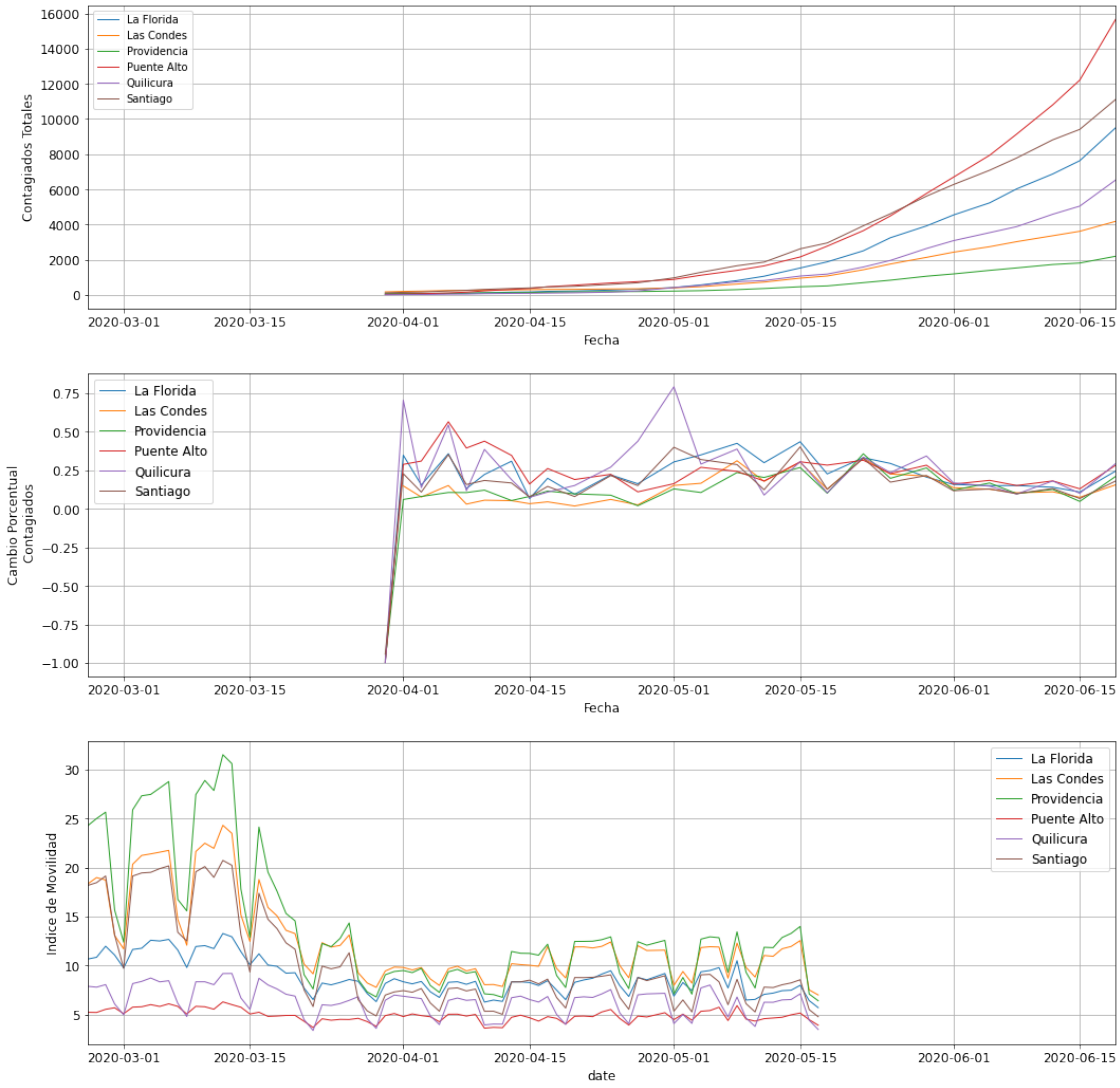
[3]:
```
def plot_contagiadosVSmov():
    """
    Realiza un gráfico triple con los contagiados totales, el porcentaje de␣
↪crecimiento de contagiados y
    e índice de movilidad para todas las comunas que se encuentren en la base de␣
↪datos.
    :return: realiza un gráfico
    """
    fig, ax = plt.subplots(3)
    fig.tight_layout(pad=3.0)
    query = (TimeSerie.
             select(TimeSerie.date, TimeSerie.value, TimeSerie.comuna_id, Comuna.
↪comuna_name).
             join(Comuna, on=(TimeSerie.comuna_id == Comuna.comuna_id)).
             where(TimeSerie.serie_id == 0))
    df_c = pd.DataFrame(list(query.dicts()))
    df_c.set_index('date', inplace=True)
    df_c.groupby('comuna_name')['value'].plot(ax=ax[0], legend=True, linewidth=1)

    df_c['pct_change'] = df_c['value'].pct_change()
    df_c.groupby('comuna_name')['pct_change'].plot(ax=ax[1], legend=True,␣
↪linewidth=1)

    query = (TimeSerie.
             select(TimeSerie.date, TimeSerie.value, TimeSerie.comuna_id, Comuna.
↪comuna_name).
             join(Comuna, on=(TimeSerie.comuna_id == Comuna.comuna_id)).
             where(TimeSerie.serie_id == 1))
    df_m = pd.DataFrame(list(query.dicts()))
    df_m.set_index('date', inplace=True)
    df_m.groupby('comuna_name')['value'].plot(ax=ax[2], legend=True, linewidth=1)

    ax[0].set_xlim(min(min(df_m.index), min(df_c.index)), max(max(df_c.index),␣
↪max(df_m.index)))
    ax[1].set_xlim(min(min(df_m.index), min(df_c.index)), max(max(df_c.index),␣
↪max(df_m.index)))
    ax[2].set_xlim(min(min(df_m.index), min(df_c.index)), max(max(df_c.index),␣
↪max(df_m.index)))
    ax[0].set_ylabel("Contagiados Totales")
    ax[1].set_ylabel("Cambio Porcentual\n Contagiados")
    ax[2].set_ylabel("Indice de Movilidad")
    ax[0].set_xlabel("Fecha")
    ax[1].set_xlabel("Fecha")
    ax[0].legend(prop={'size': 10})
```

```
    ax[0].grid()
    ax[1].grid()
    ax[2].grid()
    plt.show()

def quarantine_analisis(df_c, df_q, df_m, df_s, name):
    """
    Utiliza los dataframes de contagiados, cuarentena, movilidad y casos nuevos␣
 ↪por inicio de síntomas para una sóla comuna
    y realiza graficos comparativos en base a estos datos.
    :param df_c: dataframe de contagiados de la comuna
    :param df_q: dataframe de cuarentenas de la comuna
    :param df_m: dataframe de indice de movilidad de la comuna
    :param df_s: dataframe de casos nuevos pro inicio de síntomas de la comuna
    :param name: nombre de la comuna que se está analizando
    :return: realiza un gráfico.
    """
    fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, 1)
    fig.suptitle("Análisis Cuarentena " + name)
    fig.tight_layout(pad=4)
    df_r = df_c.pct_change()

    df_c['value'].plot(linewidth=1, label=name, ax=ax1)
    df_r['value'].plot(linewidth=1, label=name, ax=ax2)
    df_m['value'].plot(linewidth=1, label=name, ax=ax3)
    df_s['value'].plot(linewidth=1, marker='o', label=name, ax=ax4)

    for quarantine in df_q['init_day']:
        ax1.axvline(quarantine, color='lightcoral', linestyle='-', linewidth=3,␣
 ↪label='Inicio Cuarentena')
        ax2.axvline(quarantine, color='lightcoral', linestyle='-', linewidth=3)
        ax3.axvline(quarantine, color='lightcoral', linestyle='-', linewidth=3)
        ax4.axvline(quarantine, color='lightcoral', linestyle='-', linewidth=3)

    for quarantine in df_q['end_day']:
        ax1.axvline(quarantine, color='forestgreen', linestyle='--',␣
 ↪linewidth=2, label='Fin Cuarentena')
        ax2.axvline(quarantine, color='forestgreen', linestyle='--', linewidth=2)
        ax3.axvline(quarantine, color='forestgreen', linestyle='--', linewidth=2)
        ax4.axvline(quarantine, color='forestgreen', linestyle='--', linewidth=2)

    # Seteamos parámetros para que el gráfico sea más bonito :P
    ax1.set_xlim(min(min(df_c.index), min(df_m.index), min(df_s.index)),
                 max(max(df_c.index), max(df_m.index), max(df_s.index)))

    ax1.tick_params(labelbottom=False)
    ax1.set_xlabel('')
```

```
    ax1.grid()
    ax2.set_xlim(min(min(df_c.index), min(df_m.index), min(df_s.index)),
                 max(max(df_c.index), max(df_m.index), max(df_s.index)))
    ax2.tick_params(labelbottom=False)
    ax2.set_xlabel('')
    ax2.grid()
    ax3.set_xlim(min(min(df_c.index), min(df_m.index), min(df_s.index)),
                 max(max(df_c.index), max(df_m.index), max(df_s.index)))
    ax3.set_xlabel('')
    ax3.grid()
    ax4.set_xlim(min(min(df_c.index), min(df_m.index), min(df_s.index)),
                 max(max(df_c.index), max(df_m.index), max(df_s.index)))
    ax4.grid()
    ax4.set_xlabel('Fecha')
    ax1.set_ylabel("Contagiados \nTotales")
    ax2.set_ylabel("Cambio Porcentual\n Contagiados")
    ax3.set_ylabel("Indice de\n Movilidad")
    ax4.set_ylabel("Casos Nuevos por\n fecha de inicio síntomas")
    plt.show()
```

En primer lugar, mostramos una perspectiva general de la pandemia para algunas comunas. Por simp

[4]:
```
plt.rcParams['figure.figsize'] = [15, 15]
plt.rcParams.update({'font.size': 12})
plot_contagiadosVSmov()
```



Nota: Como se puede ver, ninguna de las 3 series de tiempo tiene datos completos :(

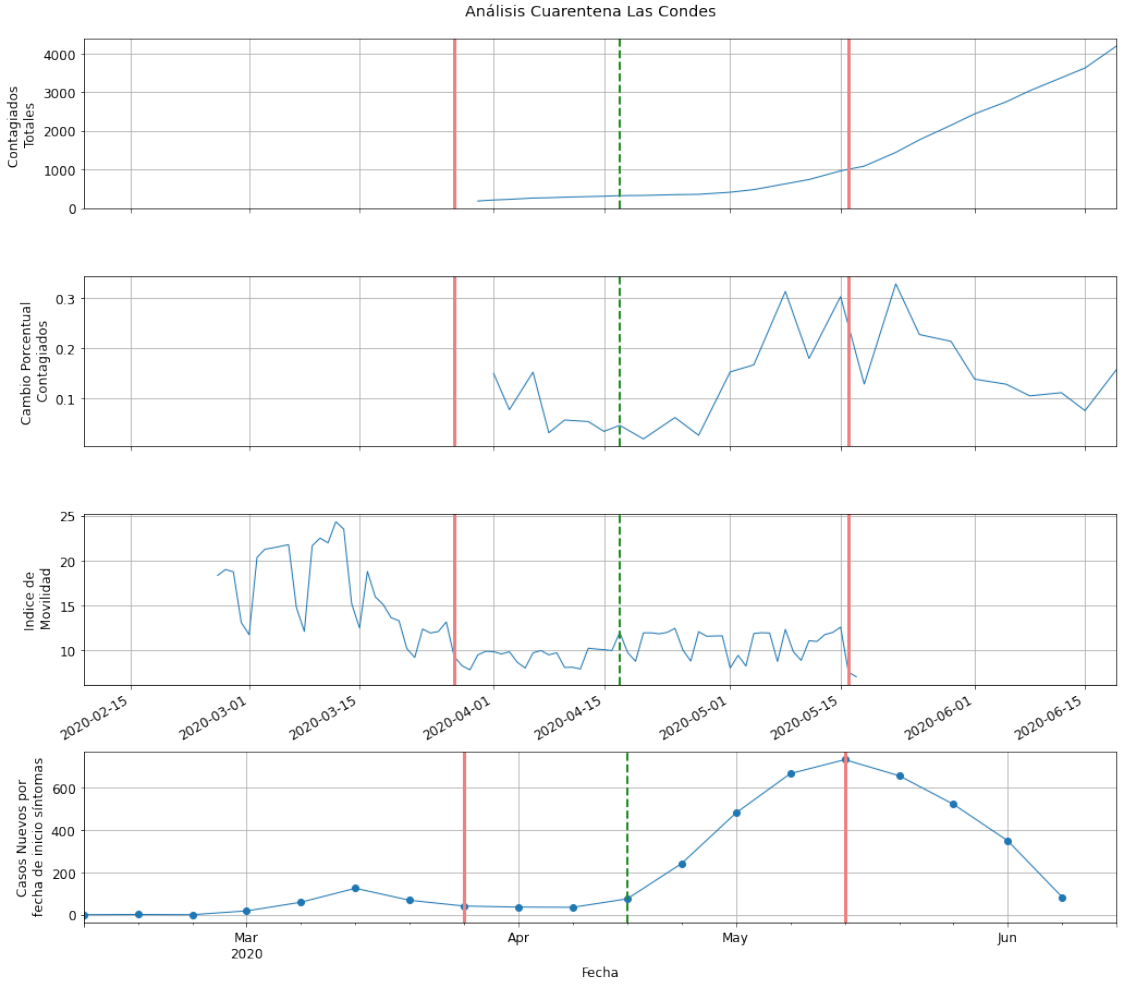
De lo anterior se puede apreciar que la movilidad disminuyó notablemente alrededor de la quincen

Por otro lado, llama la atención que si bien Providencia se mantiene con niveles relativamente a

A continuación pasaremos a analizar algunas de estas comunas para analizar si las cuarentenas fu
1. Disminución de la movilidad (Midiendo la efectividad de la medida restrictiva en si, y si efe
2. Disminución efectiva del cambio porcentual de contagiados (Midiendo la efectividad de la medi

Comenzamos analizando Las Condes, que fue una de las primeras comunas en las que se puso cuarent

[5]:
```
c_LC, q_LC, m_LC, s_LC = get_all(13114)
quarantine_analisis(c_LC, q_LC, m_LC, s_LC, "Las Condes")
print("Cuarentenas Las Condes:")
q_LC
```




```
Cuarentenas Las Condes:
```

[5]:

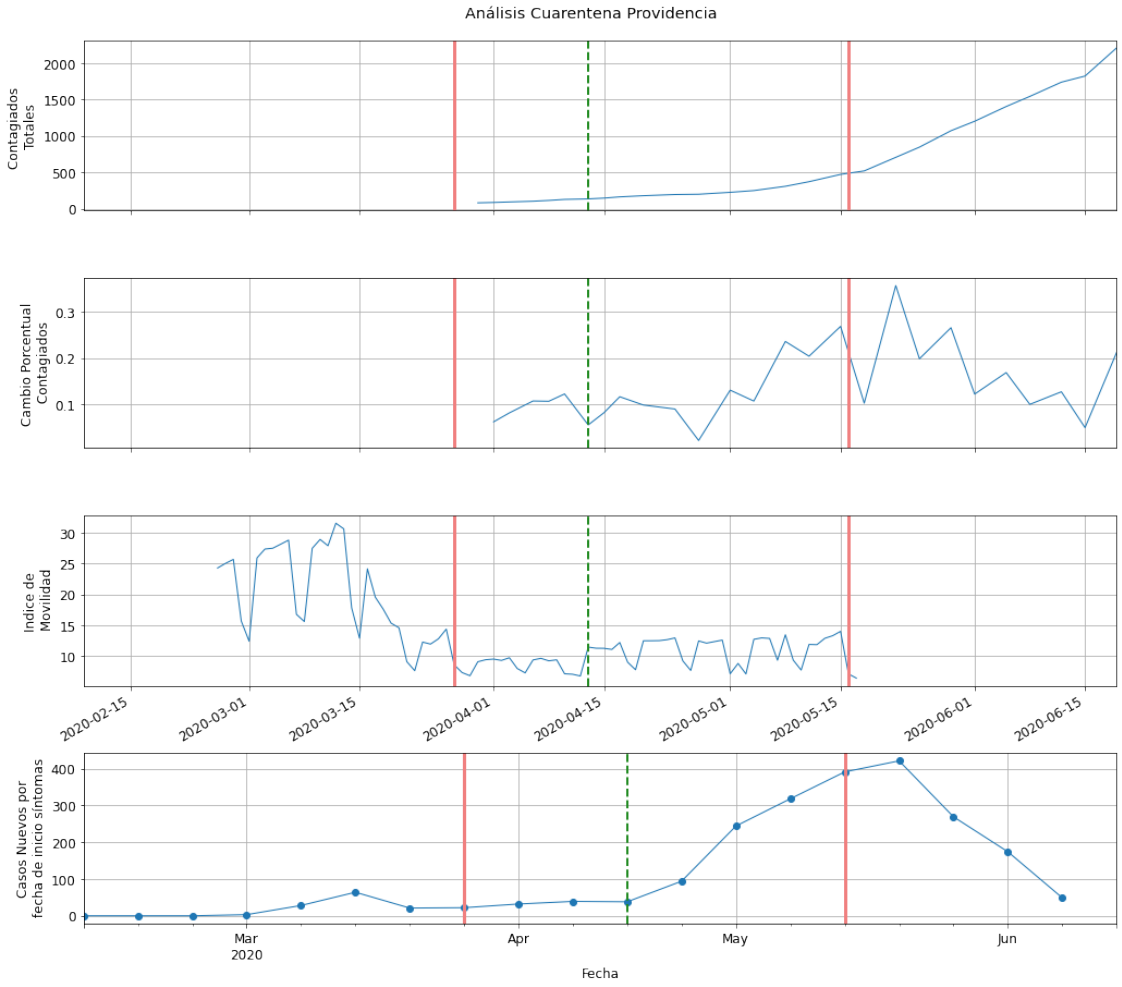
	init_day	end_day
0	2020-03-27	2020-04-17
1	2020-05-16	2020-06-27

Nota: Los Casos Nuevos por fecha de Inicio de síntomas se agrupan por "Semana Epidemiológica" por lo que no se tienen datos para cada día sino el acumulado por semana, por eso se muestran puntos sobre la línea, que representan el valor al inicio de la semana epidemiológica.

En primer lugar, notamos que, una vez que se decretó la primera cuarentena en esta comuna (27 de
Por otro lado, después de decretada esta cuarentena también se puede observar que disminuyó el a

Ahora observaremos Providencia, una de las comunas que también fue de las primeras en entrar a c

[6]:
```
c_PR, q_PR, m_PR, s_PR = get_all(13123)
quarantine_analisis(c_PR, q_PR, m_PR, s_PR, "Providencia")
print("Cuarentenas Providencia:")
q_PR
```



Cuarentenas Providencia:

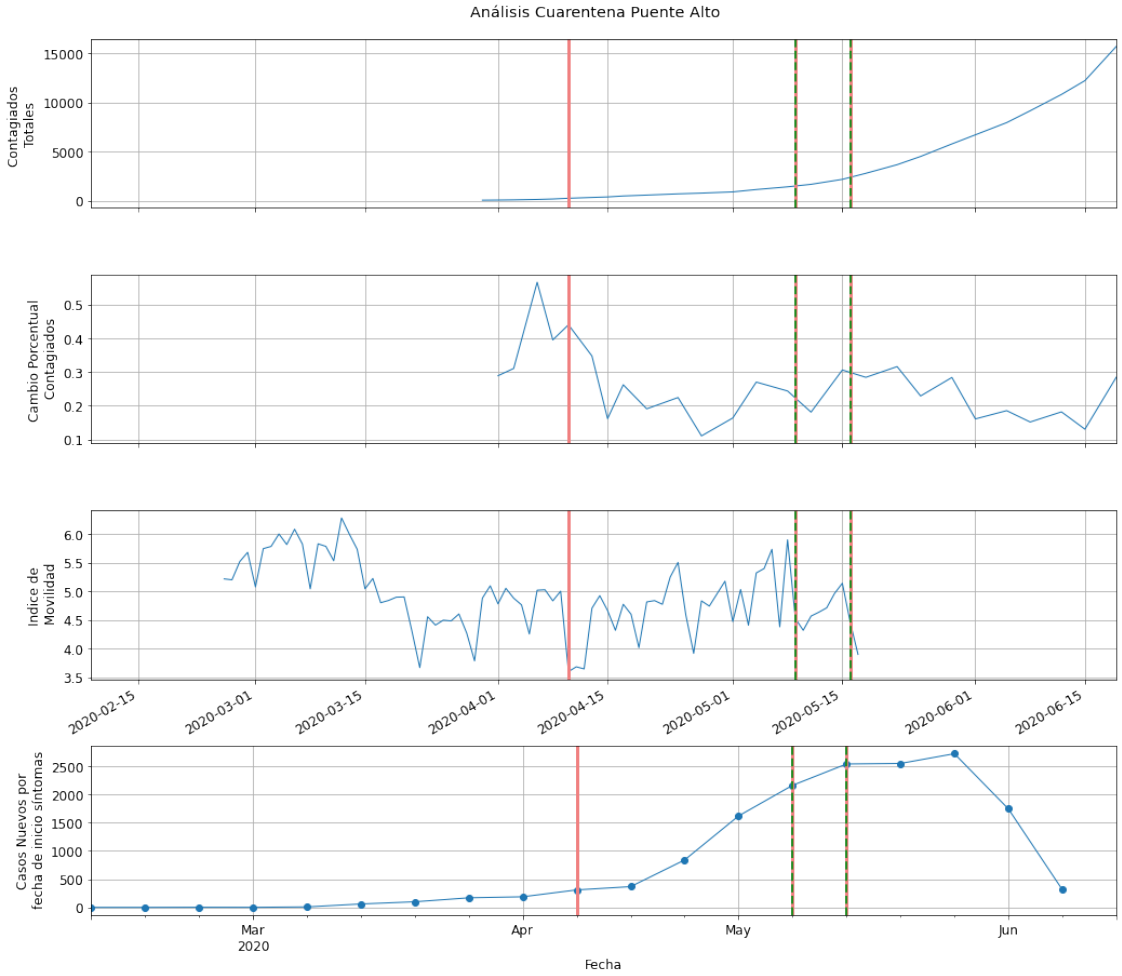
[6]:

	init_day	end_day
0	2020-03-27	2020-04-13
1	2020-05-16	2020-06-27

Como se puede ver, esta presenta las mismas cuarenetnnas que Las Condes, y un comportamiento muy

Analizaremos ahora a Puente Alto, la cual presentaba poca variación en su nivel de movilidad en

[7]:
```
c_PA, q_PA, m_PA, s_PA = get_all(13201)
quarantine_analisis(c_PA, q_PA, m_PA, s_PA, "Puente Alto")
print("Cuarentenas Puente Alto:")
q_PA
```



Cuarentenas Puente Alto:

[7]:

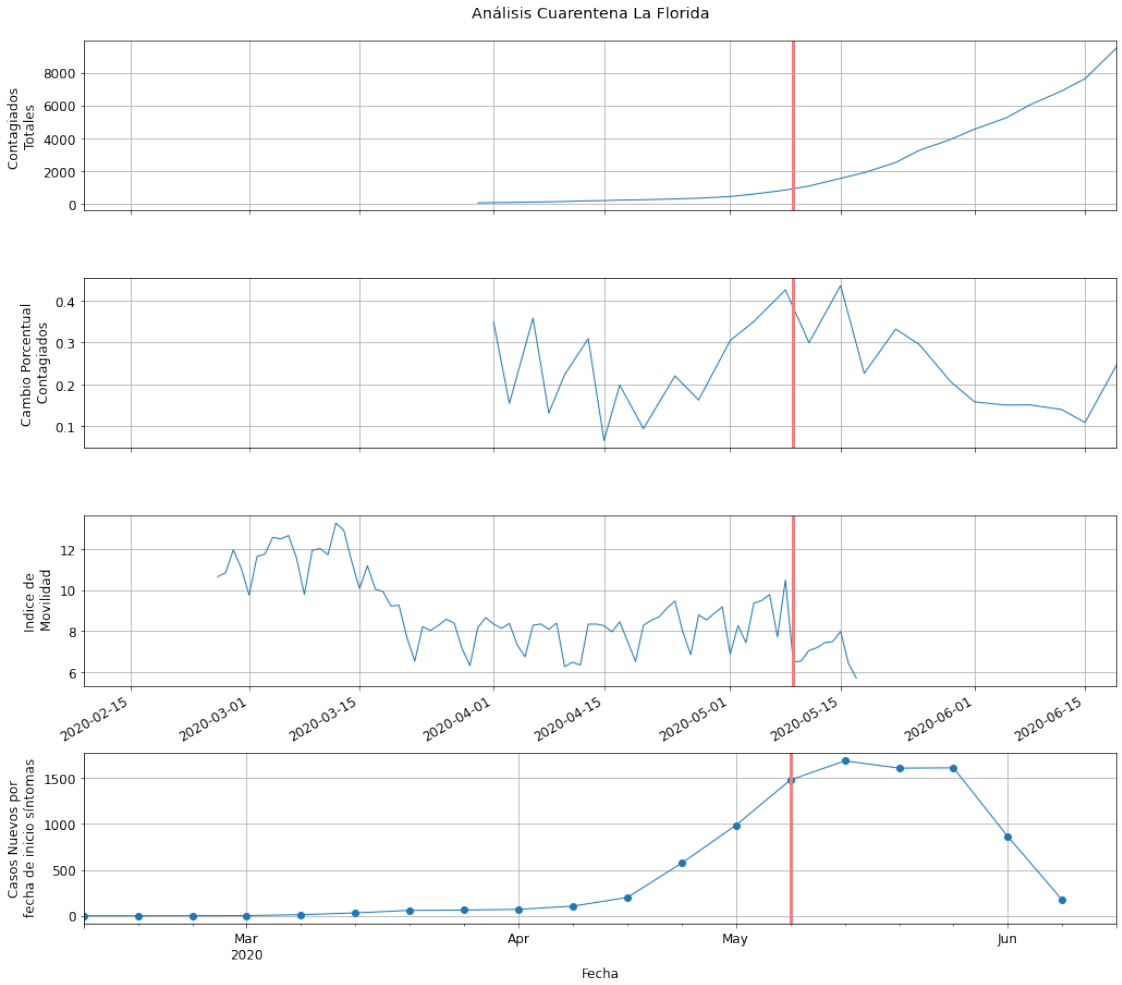
	init_day	end_day
0	2020-04-10	2020-05-09
1	2020-05-09	2020-05-16
2	2020-05-16	2020-06-27

En primer lugar, notamos que hubieron dos extensiones de cuarentena para esta comuna (representa
Esta comuna, al igual que las demás, también presenta una disminución de su índice de movilidad
Sin embargo, en lo que respecta al cambio porcentual del número de contagiados de la comuna, pod

Luego, se podría decir que si bien la cuarentena no limitó la movilidad de la comuna, si logró c

Mostramos ahora el caso de la Florida:

[8]:
```
c_LF, q_LF, m_LF, s_LF = get_all(13110)
quarantine_analisis(c_LF, q_LF, m_LF, s_LF, "La Florida")
print("Cuarentenas La Florida:")
q_LF
```



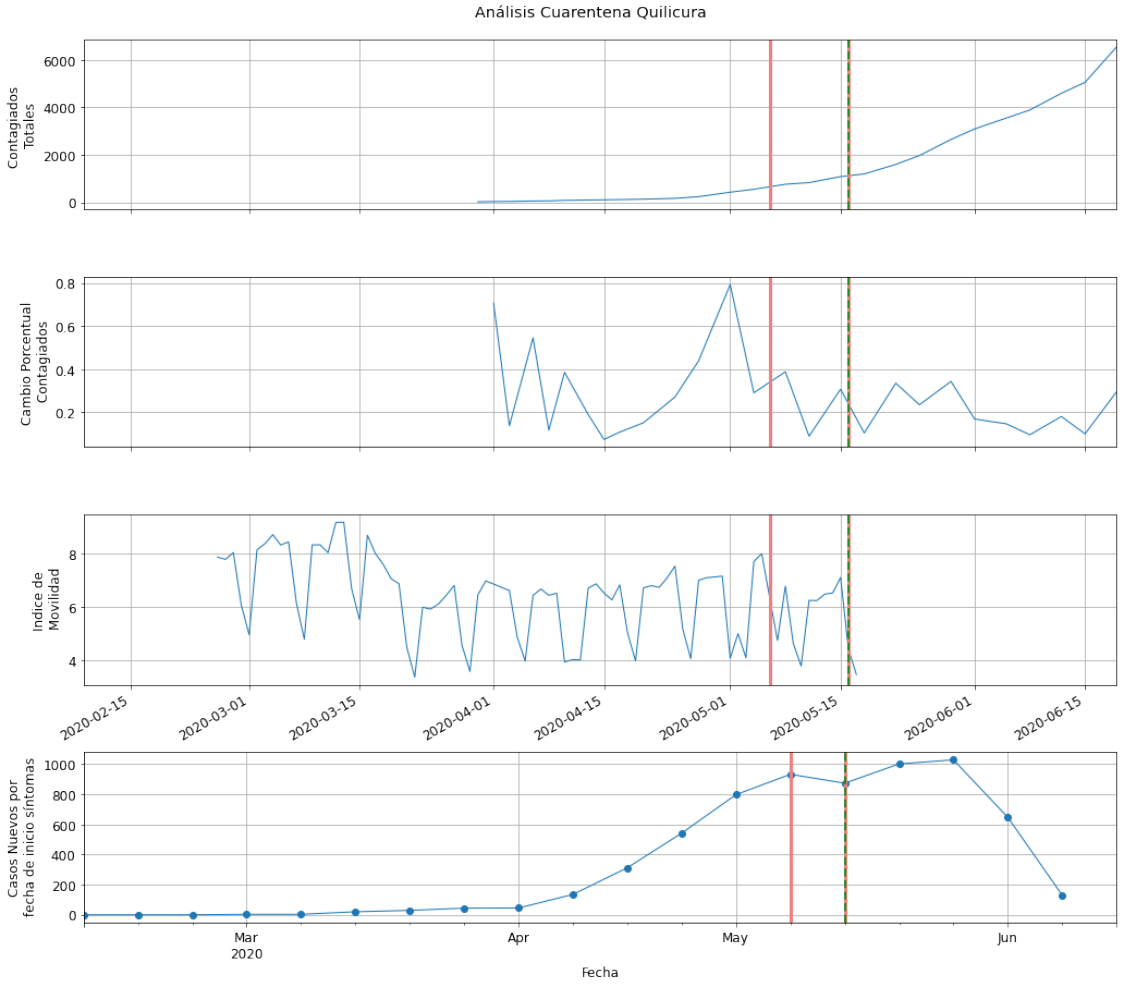
Cuarentenas La Florida:

[8]:
```
   init_day    end_day
0 2020-05-09 2020-06-27
```

En el caso de la Florida se tiene que no se decretó cuarentena oficial sino hasta el 9 de Mayo. Se puede observar también que el cambio porcentual fue aumentando con el paso de las semanas, y

Por último, analizamos el caso de Quilicura:

[9]:
```
c_QU, q_QU, m_QU, s_QU = get_all(13125)
quarantine_analisis(c_QU, q_QU, m_QU, s_QU, "Quilicura")
print("Cuarentenas Quilicura:")
q_QU
```



Cuarentenas Quilicura:

[9]:
	init_day	end_day
0	2020-05-06	2020-05-16
1	2020-05-16	2020-06-27

En el caso de Quilicura, tampoco se declaró cuarentena sino hasta el 6 de Mayo, la cuals se exte
En este caso notamos un comportamiento parecido al de Puente Alto con respecto a la movilidad, l
Por otro lado, de forma alarmante podemos notar que los primeros días de Mayo esta comuna llegó
Finalmente, al igual que en Puente Alto, también se podría concluir que si bien la movilidad no

En conclusión, podemos decir que si bien en algunas comunas como Puente Alto y Quilicura las cua