

# Seguimiento de datos COVID-19 en Colombia

Santiago Gutiérrez Orjuela  
*Escuela de Ciencias exactas e Ingeniería*  
*Universidad Sergio Arboleda - Bogotá, Colombia*  
*santiago.gutierrez02@correo.usa.edu.co*

Valeria Bermúdez Galván  
*Escuela de Ciencias exactas e Ingeniería*  
*Universidad Sergio Arboleda - Bogotá, Colombia*  
*valeria.bermudez01@correo.usa.edu.co*

Juan Sebastián Bueno Ramírez  
*Escuela de Ciencias exactas e Ingeniería*  
*Universidad Sergio Arboleda - Bogotá, Colombia*  
*juan.bueno01@correo.usa.edu.co*

## Resumen

En el presente documento se aplica la extracción de datos en internet sobre el coronavirus en Colombia para de forma selectiva presentarlos mediante diferentes gráficos. Se realiza código en Python, haciendo uso de web scraping para extraer los datos del Covid-19, se manipularán y visualizarán empleando librerías tales como pandas y matplotlib.

## Palabras clave:

Python, web scraping, datos, Covid-19.

## 1. Marco teórico

El **coronavirus** es una familia de virus que causa infecciones respiratorias que pueden ir desde el resfriado común hasta el síndrome respiratorio agudo severo. El coronavirus que se ha descubierto más recientemente causa la enfermedad por coronavirus **COVID-19**, del cuál los síntomas más habituales son fiebre, tos seca y cansancio. Para el estudio del impacto que ha causado esta enfermedad causada por el virus, se representará en gráficos la información que se tiene de los casos confirmados en el país de Colombia.

Los *datos abiertos* del gobierno de Colombia nos permitió acceder a la información asociada a los casos positivos de COVID-19, haciendo uso de web scraping en Python con sodapy que es una biblioteca que funciona como cliente en la *Socrata Open Data API*, estableciendo así conexión con la página del gobierno y obteniendo los datos necesarios, guardándolos como formato csv, estos se guardan en un dataframe con el uso de **Pandas**, el cual es una biblioteca que facilita la manipulación de estructuras de datos y nos brinda herramientas para el análisis de datos [1]. Para la visualización de los datos, se hace uso de **Matplotlib**, esta biblioteca nos permite hacer gráficas en 2D in Python, originalmente fue un emulador de los comandos de gráficos de MATLAB.

## 2. Resultados

Para el uso de *sodapy* para acceder a Socrata, escribimos las siguientes líneas, obteniendo así las 100000 primeras filas de la tabla

```
1 from sodapy import Socrata
2 import pandas as pd
3 from datetime import datetime
4 import matplotlib.pyplot as plt
5 from collections import Counter
6
7
8 client = Socrata("www.datos.gov.co", None) #Se determina la dirección de la cual se extraerán ...
    los datos
9 results = client.get("gt2j-8ykr", limit=100000) #Se obtienen los datos y se establece un límite
10
11 # Convertir a dataframe de pandas
12 results_df = pd.DataFrame.from_records(results)
```

Con el fin de usar buenas prácticas, se llenan los espacios en blanco con el campo "No definido".

```

1 df=results_df
2 df=df.fillna('No Definido') #Se llenan los datos vac os
3 tupla = (("T00:00:00.000", ""),("-", "/"))
4 for c in df.index:
5     for a, b in tupla:
6         df['fecha_de_notificaci_n'][c] = df['fecha_de_notificaci_n'][c].replace(a, b)
7         df['fis'][c] = df['fis'][c].replace(a, b)
8         df['fecha_diagnostico'][c] = df['fecha_diagnostico'][c].replace(a, b)
9         df['fecha_recuperado'][c] = df['fecha_recuperado'][c].replace(a, b)
10        df['fecha_reporte_web'][c] = df['fecha_reporte_web'][c].replace(a, b)
11        df['fecha_de_muerte'][c] = df['fecha_de_muerte'][c].replace(a, b)
12 df.head()

```

A continuación , se muestra el código usado para preparar la data y crear las gráficas necesarias.

```

1 #Se convierten los datos de string a datetime object
2 for c in df.index:
3     if(df['fecha_de_notificaci_n'][c]!='No Definido'):
4         df['fecha_de_notificaci_n'][c]=datetime.strptime(df['fecha_de_notificaci_n'][c], '%Y/%m/%d')
5     if(df['fis'][c]!='No Definido'):
6         df['fis'][c]=datetime.strptime(df['fis'][c], '%Y/%m/%d')
7     if(df['fecha_diagnostico'][c]!='No Definido'):
8         df['fecha_diagnostico'][c]=datetime.strptime(df['fecha_diagnostico'][c], '%Y/%m/%d')
9     if(df['fecha_recuperado'][c]!='No Definido'):
10        df['fecha_recuperado'][c]=datetime.strptime(df['fecha_recuperado'][c], '%Y/%m/%d')
11    if(df['fecha_reporte_web'][c]!='No Definido'):
12        df['fecha_reporte_web'][c]=datetime.strptime(df['fecha_reporte_web'][c], '%Y/%m/%d')
13    if(df['fecha_de_muerte'][c]!='No Definido'):
14        df['fecha_de_muerte'][c]=datetime.strptime(df['fecha_de_muerte'][c], '%Y/%m/%d')
15
16 x=[]
17 x=df['fecha_de_notificaci_n'] #Se llena con la fecha de notificaci n
18 c=Counter(x) #Se cuentan los datos seg n la fecha de notificaci n
19 df1 = pd.DataFrame() #Se crea una nueva referencia del DataFrame
20 df1['Fecha']=c.keys() #Se extraen los elementos nicos
21 df1['No. Casos']=c.values() #Se extrae la frecuencia de cada elemento nico
22 df1=df1.sort_values(by='Fecha') #Se ordenan seg n la fecha
23
24 x = df1['Fecha'] #Se asignan los valores al eje x
25 y = df1['No. Casos'] #Se asignan los valores al eje y
26 plt.plot(x,y) #Se grafica
27 plt.title('N mero de Casos')
28 plt.xlabel('Fecha')
29 plt.ylabel('No. Casos')
30 plt.savefig('N mero de Casos.png') #Se guarda la imagen
31
32 import matplotlib
33 import numpy as np
34
35 y=[]
36 y=df['ciudad_de_ubicaci_n'] #Se llena con la ciudad
37 d=Counter(y) #Se cuentan los datos seg n la ciudad
38 df2 = pd.DataFrame()
39 df2['Ciudad']=d.keys() #Se extraen los elementos nicos
40 df2['No. Casos']=d.values() #Se extrae la frecuencia de cada elemento nico
41 df2=df2.sort_values(by='No. Casos',ascending=False) #Se ordena seg n el n mero de casos
42
43 plt.figure(figsize=(10,7)) #Se establece el tama o de la figura
44 #Colocamos etiquetas a los ejes
45 plt.xlabel("Ciudades")
46 plt.ylabel("N mero de Casos")
47
48 #Creamos la grafica de barras utilizando 'Ciudad' como eje X y 'N mero de casos' como eje y.
49 x = df2['Ciudad'].iloc[0:6]
50 y = df2['No. Casos'].iloc[0:6]
51 plt.title("Ciudades con mayor n mero de Contagiados")
52 plt.bar(x, y)
53 #Finalmente mostramos la grafica con el metodo show()
54 plt.savefig('Ciudades.png')
55 plt.show()

```

```

56
57 tipo=[]
58 tipo=df['tipo']
59 #Se cuentan los casos seg n el tipo de contagio
60 e=Counter(tipo)
61 df3 = pd.DataFrame()
62 #Se establecen los elementos y la frecuencia de cada uno
63 df3['Tipo']=e.keys()
64 df3['No. Casos']=e.values()
65 #Se ordenan
66 df3=df3.sort_values(by='No. Casos',ascending=False)
67
68 #Se grafica el diagrama de torta
69 plt.pie(df3['No. Casos'], labels=df3['Tipo'], autopct="%0.1f %%")
70 plt.axis("equal")
71 plt.title("Casos Totales por tipo de contagio")
72 plt.savefig('TipoContagio.png')
73 #Se muestra la grafica
74 plt.show()
75
76 estado=[]
77 estado=df['estado']
78 #Se cuentan los casos seg n el estado de los pacientes
79 f=Counter(estado)
80 df4 = pd.DataFrame()
81 #Se establecen los elementos y la frecuencia de cada uno
82 df4['Estado']=f.keys()
83 df4['No. Casos']=f.values()
84 df4=df4.sort_values(by='No. Casos',ascending=False)
85
86 #Se grafica el diagrama de torta
87 plt.pie(df4['No. Casos'], labels=df4['Estado'], autopct="%0.1f %%")
88 plt.axis("equal")
89 plt.title("Estado de los pacientes")
90 plt.legend()
91 plt.savefig('EstadoPacientes.png')
92 plt.show()
93
94 pais=[]
95 pais=df['pa_s_de_procedencia']
96 #Se cuentan los casos seg n el pa s de procedencia
97 h=Counter(pais)
98 df6 = pd.DataFrame()
99 #Se establecen los elementos y la frecuencia de cada uno
100 df6['Pais de Procedencia']=h.keys()
101 df6['No. Casos']=h.values()
102 df6=df6.sort_values(by='No. Casos',ascending=False)
103 #Se hace una copia del data frame
104 df6copy = df6.iloc[1:8].copy()
105 #Se organizan seg n el n mero de casos
106 df6copy=df6copy.sort_values(by='No. Casos')
107
108 plt.figure(figsize=(9,7))
109 #Creamos la grafica pasando los valores en el eje X, Y, donde X = No de Casos y Y = Pa s de ...
110     Procedencia
111 x = df6copy['Pais de Procedencia']
112 y = df6copy['No. Casos']
113 plt.barh(x, y, align='center', alpha=0.5)
114 #a adimos una etiqueta en el eje X
115 plt.xlabel('N mero de Casos')
116 plt.title('Pa s de Procedencia')
117 plt.savefig('Pais.png')
118 plt.show()
119
120 muertes=[]
121 muertes=df['fecha_de_muerte']
122 #Se cuentan los casos seg n la fecha de muerte
123 g=Counter(muertes)
124 df5 = pd.DataFrame()
125 #Se establecen los elementos y la frecuencia de cada uno
126 df5['Muertes']=g.keys()

```

```

126 df5['No. Casos']=g.values()
127 df5=df5.sort_values(by='No. Casos',ascending=False)
128 #Se hace una copia del data frame
129 copiamuertes=df5.iloc[1:].copy()
130
131 #Se organiza seg n el n mero de muertes
132 copiamuertes=copiamuertes.sort_values(by='Muertes')
133 x = copiamuertes['Muertes']
134 y = copiamuertes['No. Casos']
135 #Se grafica
136 plt.plot(x,y)
137 plt.title('Fallecimientos ')
138 plt.xlabel('Fecha')
139 plt.ylabel('Muertes')
140 #Se guarda la gr fica como .png
141 plt.savefig('Fallecimientos.png')

```

Mediante los datos extraídos de la página del Ministerio de Colombia, se realizó la gráfica de crecimiento de los casos desde el mes de marzo hasta el mes de julio , correspondiente al número de datos extraídos. Evidenciando el aumento de los casos a lo largo de los meses.

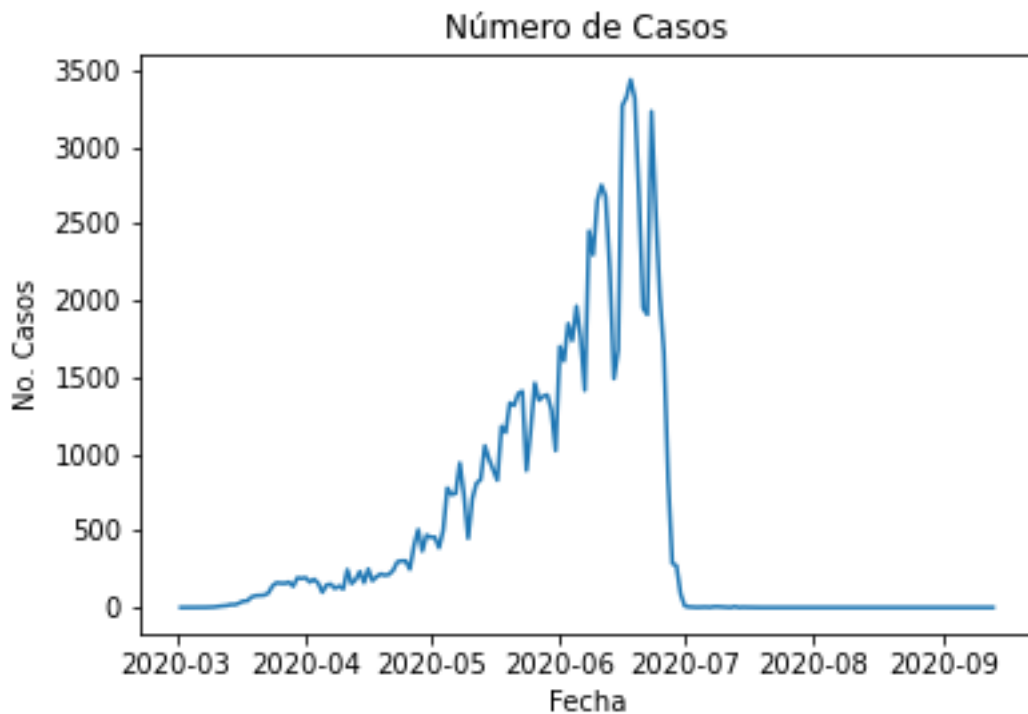


Figura 1: Número de casos positivos por meses en Colombia

En la siguiente gráfica, se muestran las ciudades con mayor número de contagios dando cuenta que Bogotá DC es la que posee mayor número de contagios.

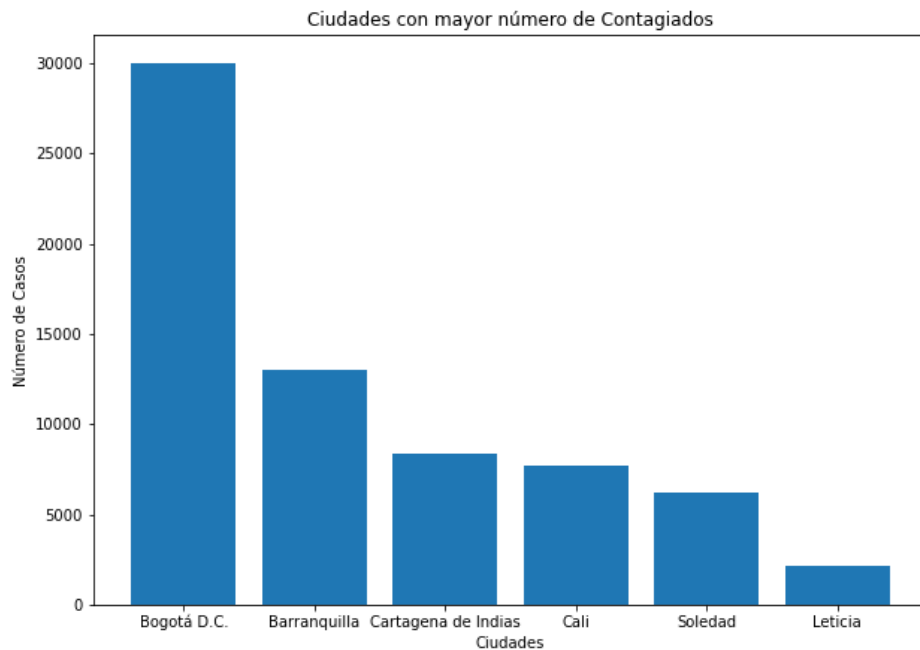


Figura 2: Número de contagios por ciudad en Colombia

Luego, se realizó una gráfica de tortas con el fin de observar el estado de los pacientes según sus síntomas, ya sea leve, asintomático, moderado, fallecido, grave, N/A permitiendo observar que el 83 % de los pacientes poseen sintomatología leve.

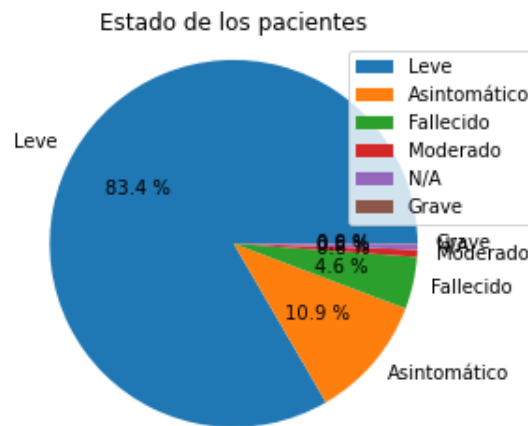


Figura 3: Estado de los pacientes de COVID-19 en Colombia

También, se realizó un gráfico de tortas para saber qué tipos de contagios poseen las personas aunque con el aumento exponencial de casos la mayoría de casos han quedado en estudio.

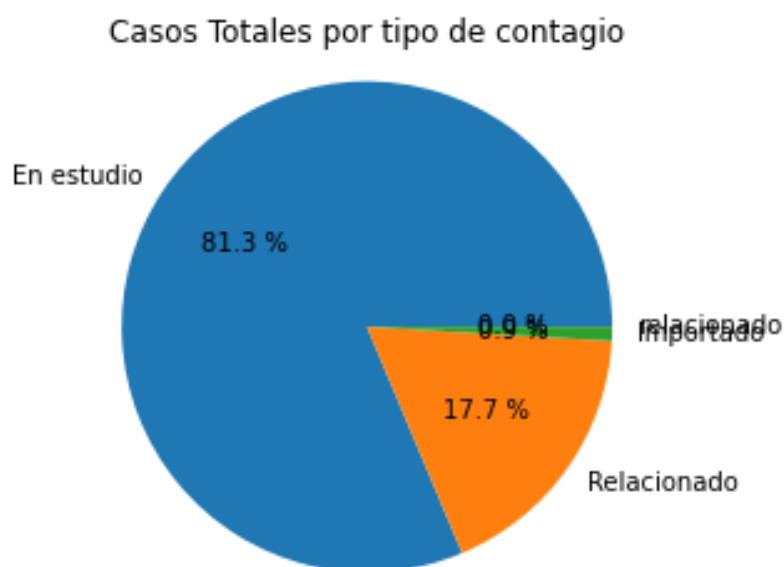


Figura 4: Tipo de contagio en pacientes colombianos

Posteriormente, se graficó el número de fallecidos por mes donde se puede evidenciar que el mes con mayor cantidad de muertos hasta el día de hoy es Julio.

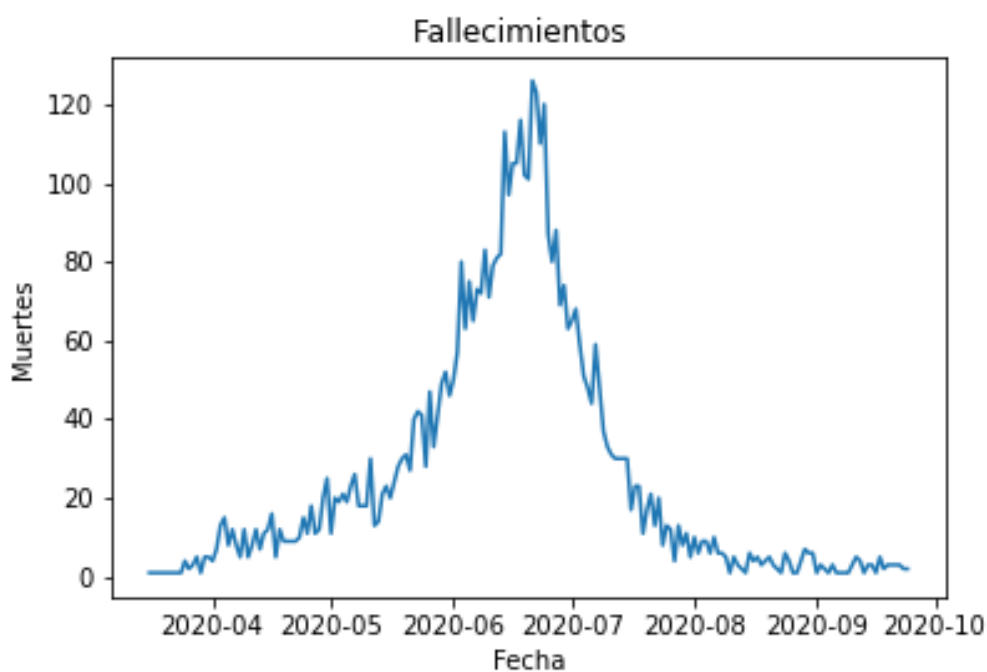


Figura 5: Número de fallecidos por mes en Colombia

Por último, mediante un gráfico de barras se mostró que la mayoría de casos importados provienen de los países España y Estados Unidos, quienes ya tuvieron su pico de la pandemia.

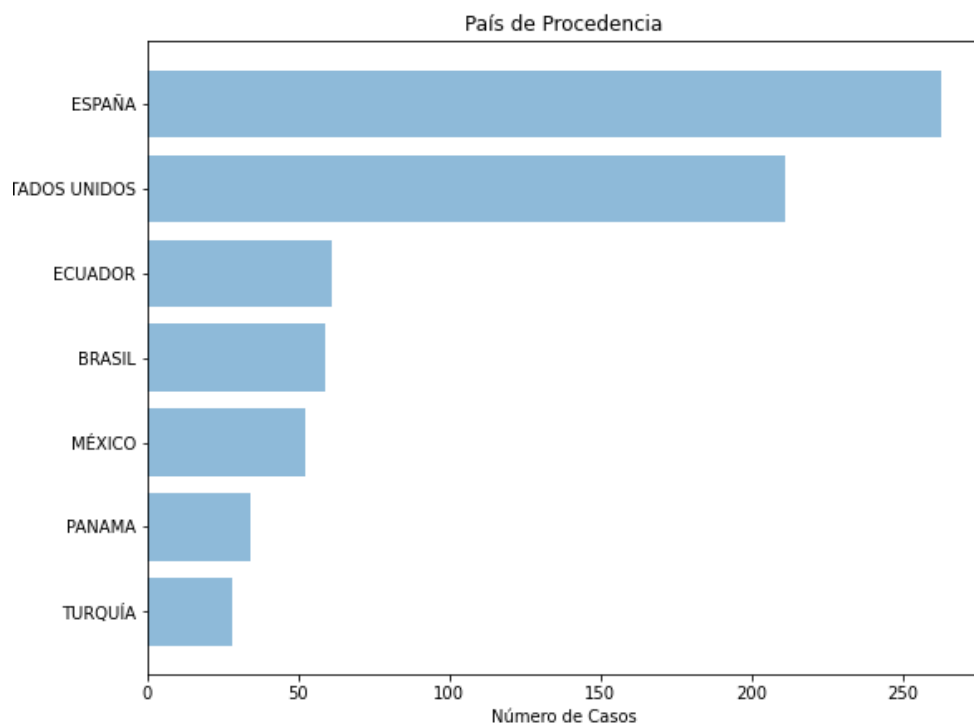


Figura 6: País de procedencia de los pacientes importados

### 3. Conclusiones

Se identifica que Pandas y Matplotlib nos facilita el manejo de los datos y su visualización.

Se evidencia que el número de contagios crece exponencialmente, tal como se ve en la figura 1.

Más del 80 % de los contagios no se sabe de dónde vienen, siguen en estudio, esto debido al crecimiento exponencial de casos no permitiendo determinar la proveniencia de la mayoría de contagios.

Bogotá es la ciudad con más contagios debido a ser la que tiene más habitantes y además donde llegan los vuelos internacionales.

### Referencias

- [1] T. pandas development team, "User guide." The pandas development team, 2008-2020.
- [2] O. M. de la Salud, "Preguntas y respuestas sobre la enfermedad por coronavirus (covid-19)." OMS, 2020.
- [3] P. S. Foundation., "Python 3.8.6 documentation." Python Software Foundation, 2001-2020.
- [4] F. D. Hunter J, Dale and the Matplotlib development team, "Tutorials." The Matplotlib development team, 2012 - 2020.
- [5] S. J., "Data visualization with plotly and pandas." Socrata, 2016.
- [6] M. de Tecnologías de la Información y las Comunicaciones, "Casos positivos de covid-19 en colombia." Datos Abiertos, 2020.