

# PROGRAMACIÓN EN PYTHON

## TRABAJO PRÁCTICO

**NOMBRE:** DIEGO TAMAYO URGILÉS  
**FECHA DE ENTREGA:** 10 DE OCTUBRE DEL 2020

### 1. Descripción general del trabajo

Este trabajo sirvió para aplicar y reforzar los conocimientos adquiridos en las clases virtuales del curso introductorio. Se revisaron varios tutoriales, a fin de comprender la utilidad de los paquetes numpy, pandas y matplotlib.pyplot, así como de las principales funciones que incluyen.

En este trabajo se analizó un archivo de datos de tipo csv, obtenido de la dirección <https://ourworldindata.org/coronavirus-source-data>, el cual contiene información relevante del coronavirus a nivel mundial. De todos estos datos, se extrajeron únicamente aquellos que abordan número de contagios y fallecidos ocurridos en el Ecuador, comprendidos entre el 01 de abril y el 31 de agosto del 2020.

El resultado de este análisis incluye la construcción de gráficos con información representativa sobre contagios y fallecimientos sobre la evolución del coronavirus en el Ecuador en el período analizado.

### 2. Código del programa

El código utilizado se incluye a continuación:

```
##### Programa de análisis de datos de contagios y fallecidos por el coronavirus en el Ecuador
##### para el período 01 de abril del 2020 al 31 de agosto del 2020
##### Autor: Diego Tamayo U.
##### Fuente de los datos: https://ourworldindata.org/coronavirus-source-data
##### Tipo de archivo analizado: Comma Separated Values (csv)
##### Salidas del programa: 1) Figura con 4 subplots para mostrar: el número de casos diarios,
##### el número de fallecidos diarios,
##### el número acumulado de contagiados y el número acumulado de fallecidos
##### 2) Figura con 2 subplots para mostrar: un gráfico de barras comparativo con el número
##### de contagiados y fallecidos en un mismo día y un gráfico de barras comparativo para
##### mostrar el número acumulado por millón de habitantes del número de contagios y el
##### número de fallecidos.

# Importamos las librería numpy, pandas y matplotlib.pyplot
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```

# Leemos el archivo csv y lo almacenamos en un dataframe
data = pd.read_csv("owid-covid-data.csv")

# Se reemplazan los espacios vacíos en los nombres con '_'
data.columns =[column.replace(" ", "_") for column in data.columns]

# Se obtiene un nuevo dataframe (df_ecuador) que contenga solo las filas correspondientes a Ecuador de un determinado rango de fechas.
df_ecuador = data[(data.location == "Ecuador") & (data.date >= "2020-04-01")]
& (data.date <= "2020-08-31")]
df_ecuador

# Se filtran únicamente las columnas relevantes para el análisis y, se guarda el resultado en un nuevo dataframe (df_final).
df_final = df_ecuador[["location", "date", "total_cases", "new_cases",
"total_deaths", "new_deaths", "total_cases_per_million",
"total_deaths_per_million"]]
df_final

# Recalculamos el índice para que sea secuencial
df_final.index = [x for x in range(0, len(df_final.values))]
df_final

# Calculamos el número de filas del dataframe df_final
filas = len(df_final.axes[0])

# Calculamos el número de columnas del dataframe df_final
columnas = len(df_final.axes[1])

# Imprimimos el número de filas y columnas del dataframe df_final
print("Número de filas: ", filas)
print("Número de columnas: ", columnas)

# Configuraciones sobre el objeto de la figura (figure)
# Mostramos la configuración inicial
print(plt.rcParams.get('figure.figsize'))

# Aplicamos algunos cambios sobre la configuración inicial
fig_size = plt.rcParams["figure.figsize"]
fig_size[0] = 15
fig_size[1] = 8
plt.rcParams["figure.figsize"] = fig_size

# Mostramos la configuración obtenida luego del cambio
print(plt.rcParams.get('figure.figsize'))

# Filtramos el dataframe df_final, para que solo nos muestre los valores de los eventos obtenidos cada 10 días

# df_final_filtrado es el dataframe base que se utilizará en la construcción de los gráficos de las dos salidas del programa
df_final_filtrado = df_final[(df_final.index % 10 == 0)]

```

```

# Salida uno del programa: gráfico con 4 subplots
# Recordamos el contenido del dataframe que será utilizado para los gráficos
df_final_filtrado

# Definimos un objeto figure, que contendrá 4 subplots
fig, a = plt.subplots(2,2)

# Establecemos la altura del gráfico
fig.set_figheight(12)
fig.suptitle("Coronavirus en el Ecuador (01 de abril al 31 de agosto del 2020)", fontsize=20, fontweight='bold', color="blue", y=0.98)

# Definimos los parámetros del primer subplot, ubicado en la primera fila y primera columna, en la posición a[0][0]
a[0][0].plot(df_final_filtrado["date"], df_final_filtrado["total_cases"], color='green')
a[0][0].set_title('Total de contagios', fontsize=16, fontweight='bold')
a[0][0].set_ylabel('No. total de contagios', fontsize=12)

# Cambiamos la orientación de las etiquetas para su correcta visualización
a[0][0].tick_params(axis='x', labelrotation=90)

# Mostramos una cuadrícula para mejor lectura de los datos
a[0][0].grid()

# Definimos los parámetros del segundo subplot, ubicado en la primera fila y segunda columna, en la posición a[0][1]
a[0][1].plot(df_final_filtrado["date"], df_final_filtrado["new_cases"], color='red')
a[0][1].set_title('Nuevos contagios', fontsize=16, fontweight='bold')
a[0][1].set_ylabel('No. de nuevos contagios', fontsize=12)

# Cambiamos la orientación de las etiquetas para su correcta visualización
a[0][1].tick_params(axis='x', labelrotation=90)

# Cambiamos la orientación de las etiquetas para su correcta visualización
a[0][1].grid()

# Definimos los parámetros del tercer subplot, ubicado en la segunda fila y primera columna, en la posición a[1][0]
a[1][0].plot(df_final_filtrado["date"], df_final_filtrado["total_deaths"], color='orange')
a[1][0].set_title('Total de fallecidos', fontsize=16, fontweight='bold')
a[1][0].set_ylabel('No. total de fallecidos', fontsize=12)

# Cambiamos la orientación de las etiquetas para su correcta visualización
a[1][0].tick_params(axis='x', labelrotation=90)

# Mostramos una cuadrícula para mejor lectura de los datos
a[1][0].grid()

# Definimos los parámetros del cuarto subplot, ubicado en la segunda fila y segunda columna, en la posición a[1][1]

```

```

a[1][1].plot(df_final_filtrado["date"], df_final_filtrado["new_deaths"],
color='blue')
a[1][1].set_title('Nuevos fallecidos', fontsize=16, fontweight='bold')
a[1][1].set_ylabel('No. total de nuevos fallecidos', fontsize=12)

# Cambiamos la orientación de las etiquetas para su correcta visualización
a[1][1].tick_params(axis='x', labelrotation=90)

# Mostramos una cuadricula para mejor lectura de los datos
a[1][1].grid()

# Establecemos con figuraciones adicionales para mejorar la apariencia visual de los gráficos
plt.subplots_adjust(left=0.125,
bottom=0.1,
right=0.9,
top=0.9,
wspace=0.2,
hspace=0.4)

# Mostramos el gráfico que contiene los 4 subplots
plt.show()

# Salida dos del programa: gráfico con 2 subplots
# Recordamos el contenido del dataframe que será utilizado para los gráficos
df_final_filtrado

# Definimos un nuevo objeto de tipo figure, el cual contiene 2 subplots
fig, (ax0, ax1) = plt.subplots(nrows=1, ncols=2)

# Definimos la altura del gráfico
fig.set_figheight(5)
fig.suptitle("Coronavirus en el Ecuador (01 de abril al 31 de agosto del 2020)", fontsize=20, fontweight='bold', color="blue", y=0.98)

# Establecemos el valor del ancho de cada barra que se muestre en los gráficos
ancho_barra = 5.00

# Definimos los dataframes que se utilizarán
contagiados = df_final_filtrado["new_cases"]
fallecidos = df_final_filtrado["new_deaths"]
total_casos_por_millon = df_final_filtrado["total_cases_per_million"]
total_fallecidos_por_millon = df_final_filtrado["total_deaths_per_million"]

# Definimos los parámetros del primer subplot, ubicado en la primera fila y primera columna, en la posición a[0][0] = ax0
ax0.bar(df_final_filtrado.index, contagiados, color="orange",
width=ancho_barra)
ax0.bar(df_final_filtrado.index + ancho_barra, fallecidos, color="purple",
width=ancho_barra)
ax0.set_title('Contagiados vs. Fallecidos', fontsize=16, fontweight='bold')
ax0.set_ylabel('No. total de nuevos fallecidos', fontsize=12)

```

```

# Mostramos una cuadrícula para mejor lectura de los datos
ax0.grid()

# Definimos los parámetros del segundo subplot, ubicado en la primera fila y primera columna,
# en la posición a[0][1] = ax1
ax1.bar(df_final_filtrado.index, total_casos_por_millon, color="green",
width=ancho_barra)
ax1.bar(df_final_filtrado.index + ancho_barra, total_fallecidos_por_millon,
color="blue", width=ancho_barra)
ax1.set_title('Contagiados vs. Fallecidos\n(acumulado por millón de
habitantes)', fontsize=16, fontweight='bold')
ax1.set_ylabel('No. total de nuevos fallecidos', fontsize=12)

# Mostramos una cuadrícula para mejor lectura de los datos
ax1.grid()

# Establecemos parámetros adicionales de configuración para mejorar la apariencia visual de
# los gráficos
plt.subplots_adjust(left=0.125,
bottom=0.1,
right=0.9,
top=0.8,
wspace=0.2,
hspace=0.8)

# Mostramos el gráfico que contiene los 2 subplots
plt.show()

# Fin del programa

```

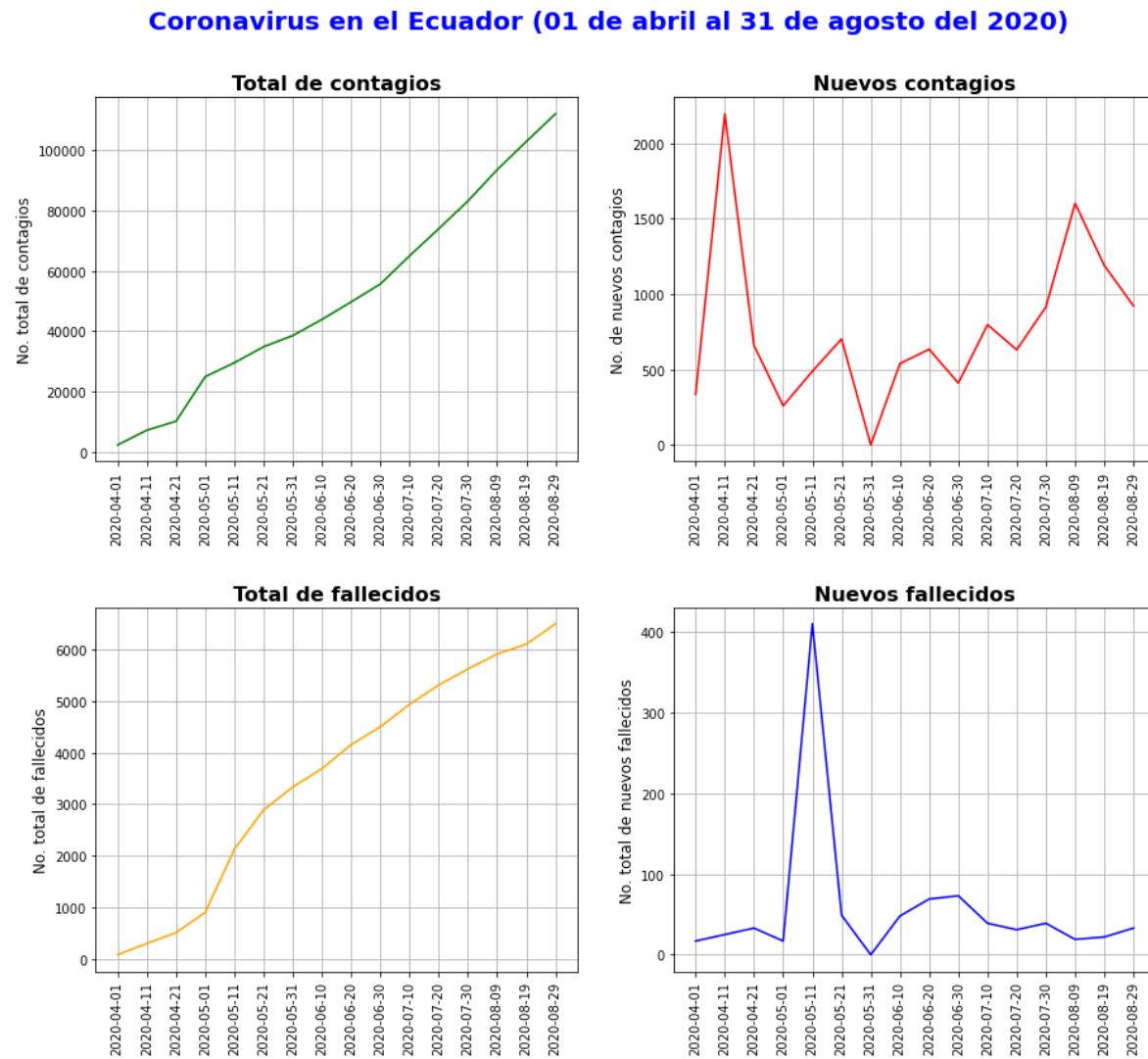
	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	total_deaths_per_million
0	Ecuador	2020-04-01	2302.0	336.0	79.0	17.0	130.476	4.478
10	Ecuador	2020-04-11	7161.0	2196.0	297.0	25.0	405.882	16.834
20	Ecuador	2020-04-21	10128.0	660.0	507.0	33.0	574.050	28.737
30	Ecuador	2020-05-01	24934.0	259.0	900.0	17.0	1413.247	51.012
40	Ecuador	2020-05-11	29559.0	488.0	2127.0	410.0	1675.390	120.557
50	Ecuador	2020-05-21	34854.0	703.0	2888.0	49.0	1975.508	163.690
60	Ecuador	2020-05-31	38571.0	0.0	3334.0	0.0	2186.185	188.969
70	Ecuador	2020-06-10	43917.0	539.0	3690.0	48.0	2489.194	209.147
80	Ecuador	2020-06-20	49731.0	634.0	4156.0	69.0	2818.729	235.560
90	Ecuador	2020-06-30	55665.0	410.0	4502.0	73.0	3155.065	255.171
100	Ecuador	2020-07-10	65018.0	797.0	4939.0	39.0	3685.188	279.940
110	Ecuador	2020-07-20	74013.0	631.0	5313.0	31.0	4195.021	301.138
120	Ecuador	2020-07-30	83193.0	914.0	5623.0	39.0	4715.338	318.709
130	Ecuador	2020-08-09	93572.0	1603.0	5916.0	19.0	5303.615	335.316
140	Ecuador	2020-08-19	102941.0	1190.0	6105.0	22.0	5834.645	346.028
150	Ecuador	2020-08-29	112141.0	922.0	6504.0	33.0	6356.097	368.644

Figura 1: Tabla utilizada en la construcción de los gráficos

### 3. Salidas del programa

La tabla que se utilizará en el programa (dataframe) es la que se muestra en la Figura 1.

La primera salida del programa es la siguiente (Figura 2):



**Figura 2: Salida uno del programa**

En estos gráficos se puede observar la marcada tendencia creciente, tanto de número de contagios como de fallecidos, considerando que es una gráfica acumulativa, podemos concluir que, de los datos analizados, aún no se observa un decrecimiento importante ni de contagios ni de fallecidos (gráficos del lado izquierdo). Por otro lado, podemos concluir que el pico de contagios en el Ecuador, ocurrió a mediados del mes de abril y que el pico en el número de fallecidos en el país, ocurrió a mediados del mes de julio (gráficos del lado derecho).

La segunda salida del programa es la siguiente (Figura 3):



Figura 3: Salida dos del programa

En estos dos gráficos, se muestra una comparativa entre el número de contagios y el número de fallecidos. Se observa en el primer gráfico, que luego de haber transcurrido cerca de 40 días desde el 01 de abril, el número de fallecidos creció considerablemente (barras de color morado), luego de 20 días de haberse presentado el pico máximo de contagios en el país (barras de color amarillo). En el segundo gráfico se observa que el número de contagios sigue creciendo de forma constante (barras de color verde), pero el número de fallecidos, aunque sigue produciéndose, parece haber disminuido en ritmo de ocurrencia, a lo largo del período analizado (barras de color azul).

#### 4. Conclusiones

Este trabajo me dio la oportunidad de aprender un nuevo lenguaje de programación, orientado a análisis de datos, el cual desde ahora se constituye en una poderosa herramienta para utilizarla en los trabajos e investigaciones venideros.