

# Analisis

## Importacion de librerias a usar.

```
In [1]: ▶ 1 import pandas as pd
          2 import numpy as np
          3 from datetime import datetime, timedelta
          4 from sklearn.metrics import mean_squared_error
          5 from scipy.optimize import curve_fit
          6 from scipy.optimize import fsolve
          7 from sklearn import linear_model
          8 import matplotlib.pyplot as plt
          9 %matplotlib inline
```

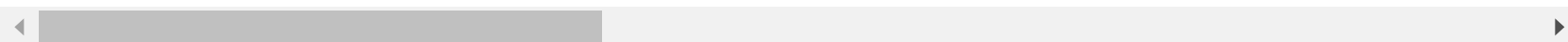
## Lectura del archivo csv actualizado a 2021

```
In [2]: 1 url = 'http://cowid.netlify.com/data/owid-covid-data.csv'
        2
        3 df = pd.read_csv(url)
        4 df
```

Out[2]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths
0	AFG	Asia	Afghanistan	2020-02-24	1.0	1.0	NaN	NaN	NaN	NaN
1	AFG	Asia	Afghanistan	2020-02-25	1.0	0.0	NaN	NaN	NaN	NaN
2	AFG	Asia	Afghanistan	2020-02-26	1.0	0.0	NaN	NaN	NaN	NaN
3	AFG	Asia	Afghanistan	2020-02-27	1.0	0.0	NaN	NaN	NaN	NaN
4	AFG	Asia	Afghanistan	2020-02-28	1.0	0.0	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...
101743	ZWE	Africa	Zimbabwe	2021-07-07	60227.0	2264.0	1480.429	1973.0	34.0	34.0
101744	ZWE	Africa	Zimbabwe	2021-07-08	62383.0	2156.0	1594.571	2029.0	56.0	56.0
101745	ZWE	Africa	Zimbabwe	2021-07-09	65066.0	2683.0	1771.857	2084.0	55.0	55.0
101746	ZWE	Africa	Zimbabwe	2021-07-10	66853.0	1787.0	1884.000	2126.0	42.0	42.0
101747	ZWE	Africa	Zimbabwe	2021-07-11	67765.0	912.0	1898.714	2185.0	59.0	59.0

101748 rows × 60 columns



**Imprimos los resultados y agregamos el numero del dia**

```
In [3]: 1 df = df[df['location'].isin(['Ecuador'])] #Filtro la Informacion solo para Ecuador
2 df = df.loc[:,['date','total_cases']] #Selecciono las columnas de analisis
3 # Expresar Las fechas en numero de dias desde el 01 Enero
4 FMT = '%Y-%m-%d'
5 date = df['date']
6 df['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("2020-01-01", FMT)).days)
7
8 df
```

Out[3]:

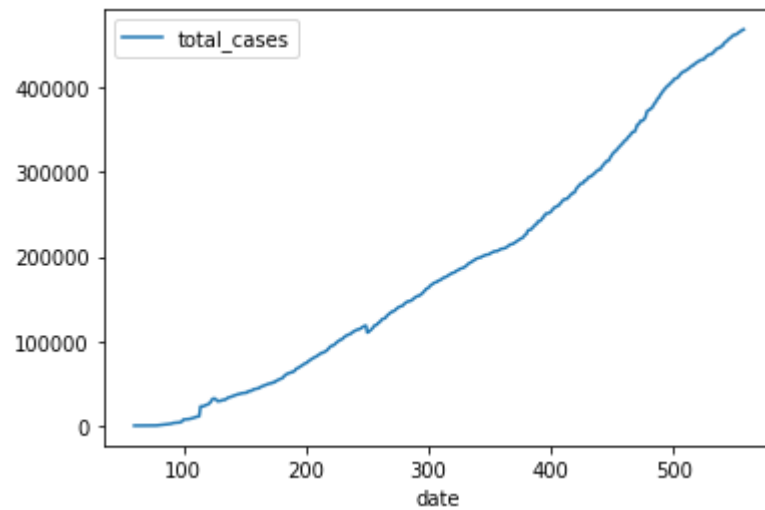
	date	total_cases
26640	60	6.0
26641	61	6.0
26642	62	7.0
26643	63	10.0
26644	64	13.0
...	...	...
27133	553	465029.0
27134	554	465878.0
27135	555	467073.0
27136	556	467976.0
27137	557	468346.0

498 rows × 2 columns

## Total de casos

```
In [4]: 1 df.plot(x='date', y='total_cases')
```

```
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1ded8f4bb48>
```



## Modelo lineal

```
In [5]: ▶ 1 x = list(df.iloc[:, 0]) # Fecha
          2 y = list(df.iloc[:, 1]) # Numero de casos
          3 # Creamos el objeto de Regresión Lineal
          4 regr = linear_model.LinearRegression()
          5
          6 # Entrenamos nuestro modelo
          7 regr.fit(np.array(x).reshape(-1, 1), y)
          8
          9 # Veamos Los coeficientes obtenidos, En nuestro caso, serán La Tangente
         10 print('Coefficients: \n', regr.coef_)
         11 # Este es el valor donde corta el eje Y (en X=0)
         12 print('Independent term: \n', regr.intercept_)
         13 # Error Cuadrado Medio
```

Coefficients:

[973.61259583]

Independent term:

-111601.4657317187

```
In [6]: ▶ 1 #Vamos a comprobar:
          2 # Quiero predecir cuántos "Casos" voy a obtener por en el día 100,
          3 # según nuestro modelo, hacemos:
          4 y_prediccion = regr.predict([[100]])
          5 print(int(y_prediccion))
```

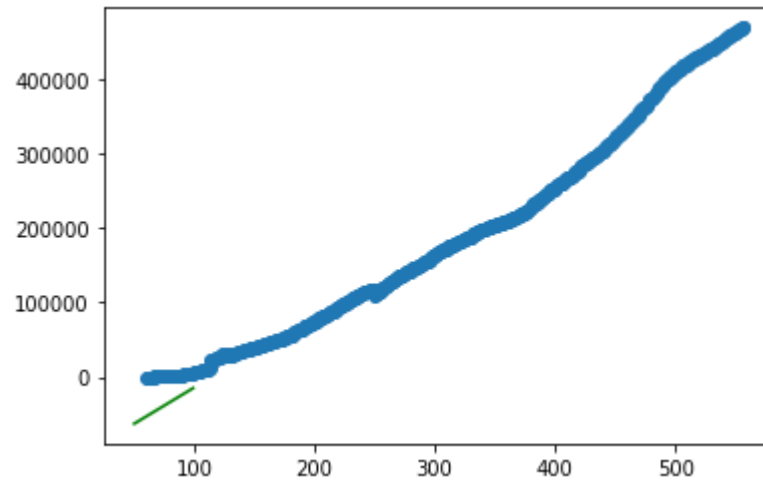
-14240

## Grafica del Modelo Lineal

In [7]:

```
1 #Graficar
2 plt.scatter(x, y)
3 x_real = np.array(range(50, 100))
4 print(x_real)
5 plt.plot(x_real, regr.predict(x_real.reshape(-1, 1)), color='green')
6 plt.show()
```

```
[50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73
74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97
98 99]
```

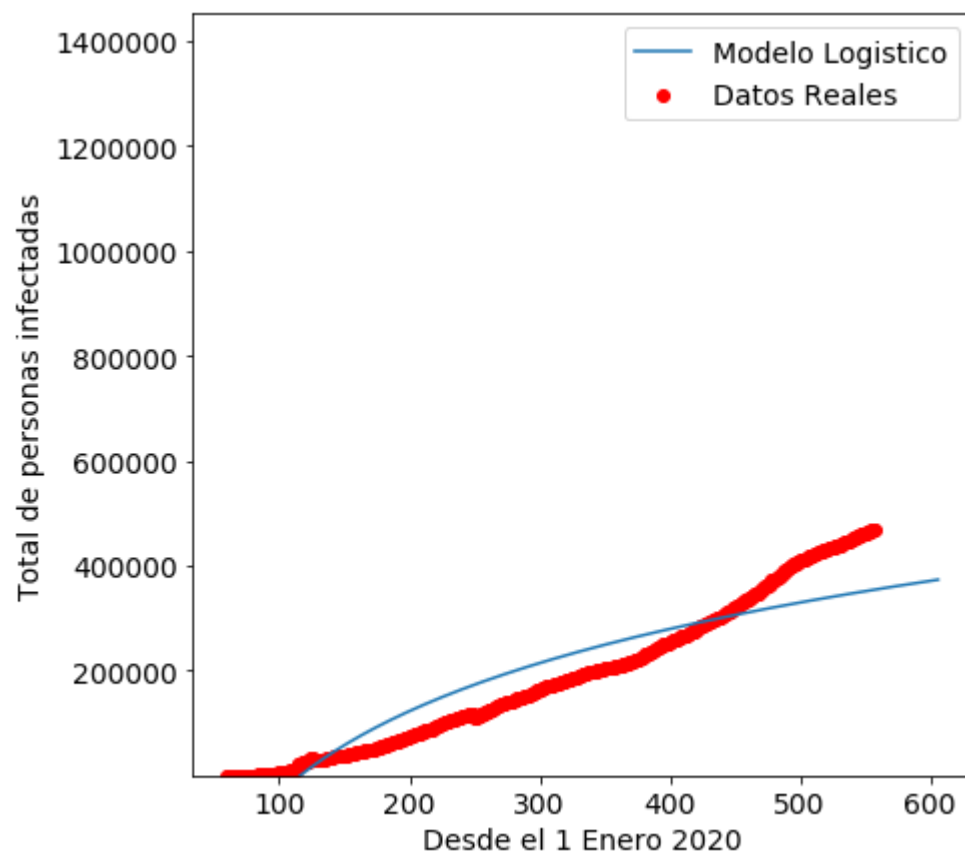


## Modelo Logistico

```
In [8]: ▶ 1 def modelo_logistico(x,a,b):  
2         return a+b*np.log(x)  
3  
4 exp_fit = curve_fit(modelo_logistico,x,y) #Extraemos los valores de Los paramatros  
5 print(exp_fit)  
  
(array([-1073930.84914347,  225851.44140357]), array([[ 6.28386604e+08, -1.11221073e+08],  
          [-1.11221073e+08,  1.98936097e+07]]))
```

## Grafica del Modelo Logistico

```
In [9]: ▶ 1 pred_x = list(range(min(x),max(x)+50)) # Predecir 50 dias mas
2 plt.rcParams['figure.figsize'] = [7, 7]
3 plt.rc('font', size=14)
4 # Real data
5 plt.scatter(x,y,label="Datos Reales",color="red")
6 # Predicted exponential curve
7 plt.plot(pred_x, [modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i in pred_x], label="Modelo Logistico")
8 plt.legend()
9 plt.xlabel("Desde el 1 Enero 2020")
10 plt.ylabel("Total de personas infectadas")
11 plt.ylim((min(y)*0.9,max(y)*3.1)) # Definir los limites de Y
12 plt.show()
```



```
In [ ]: ▶ 1
```



