Universidad Politecnica Salesiana

Nombre: Erika Morocho

Asignatura: Simulación

Covid-19 infección en Ecuador. Modelos matemáticos y predicciones

Una comparación de modelos, lineal, polilnomico,logísticos y exponenciales aplicados a la infección por el virus Covid-19

Realizar un análisis matemático simple del crecimiento de la infección en Python y dos modelos para comprender mejor la evolución de la infección.
Se crea modelos de series temporales del número total de personas infectadas hasta la fecha (es decir, las personas realmente infectadas más las personas que han sido infectadas). Estos modelos tienen parámetros , que se estimarán por

In [5]:
```
# Importar las librerias para el analasis
import pandas as pd
import numpy as np
from datetime import datetime,timedelta
from sklearn.metrics import mean_squared_error
from scipy.optimize import curve_fit
from scipy.optimize import fsolve
from sklearn import linear_model
import matplotlib.pyplot as plt
%matplotlib inline
```

In [6]:
```
# Actualizar los datos (URL)
#https://github.com/owid/covid-19-data/tree/master/public/data
url = 'owid-covid-data.csv'
df = pd.read_csv(url)
df= df.fillna(0)
df.head()
```

Out[6]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deat
0	AFG	Asia	Afghanistan	2020-01-23	0.0	0.0	0.0	0.0	0.0	
1	AFG	Asia	Afghanistan	2020-01-24	0.0	0.0	0.0	0.0	0.0	
2	AFG	Asia	Afghanistan	2020-01-25	0.0	0.0	0.0	0.0	0.0	
3	AFG	Asia	Afghanistan	2020-01-26	0.0	0.0	0.0	0.0	0.0	
4	AFG	Asia	Afghanistan	2020-01-27	0.0	0.0	0.0	0.0	0.0	

5 rows × 50 columns

Imprimos los resultados y agregamos el numero del dia

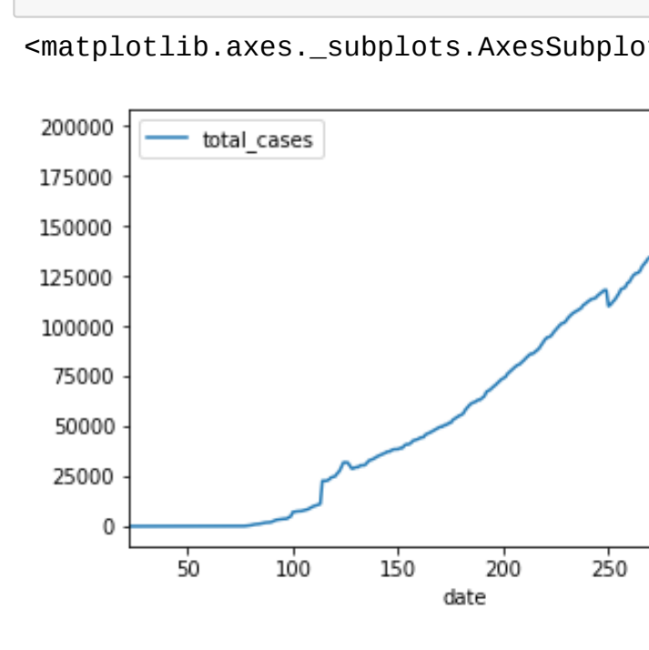
In [7]:
```
df = df[df['location'].isin(['Ecuador'])] #Filtro la Informacion solo para Ecuador
df = df.loc[:,['date','total_cases', 'new_cases']] #Selecciono las columnas de analasis
# Expresar las fechas en numero de dias desde el 01 Enero
FMT = '%Y-%m-%d'
date = df['date']
df['date'] = date.map(lambda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

Out[7]:

	date	total_cases	new_cases
16001	22	0.0	0.0
16002	23	0.0	0.0
16003	24	0.0	0.0
16004	25	0.0	0.0
16005	26	0.0	0.0

In [8]:
```
df.plot(x ='date', y='total_cases')
```

Out[8]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x18a6dcb9be0>
```



Ahora analizamos los cuatro modelos que son: la función lineal, polinomica,logística y la función exponencial . Cada modelo tiene tres parámetros , que se estimarán mediante un cálculo de ajuste de curva en los datos históricos.

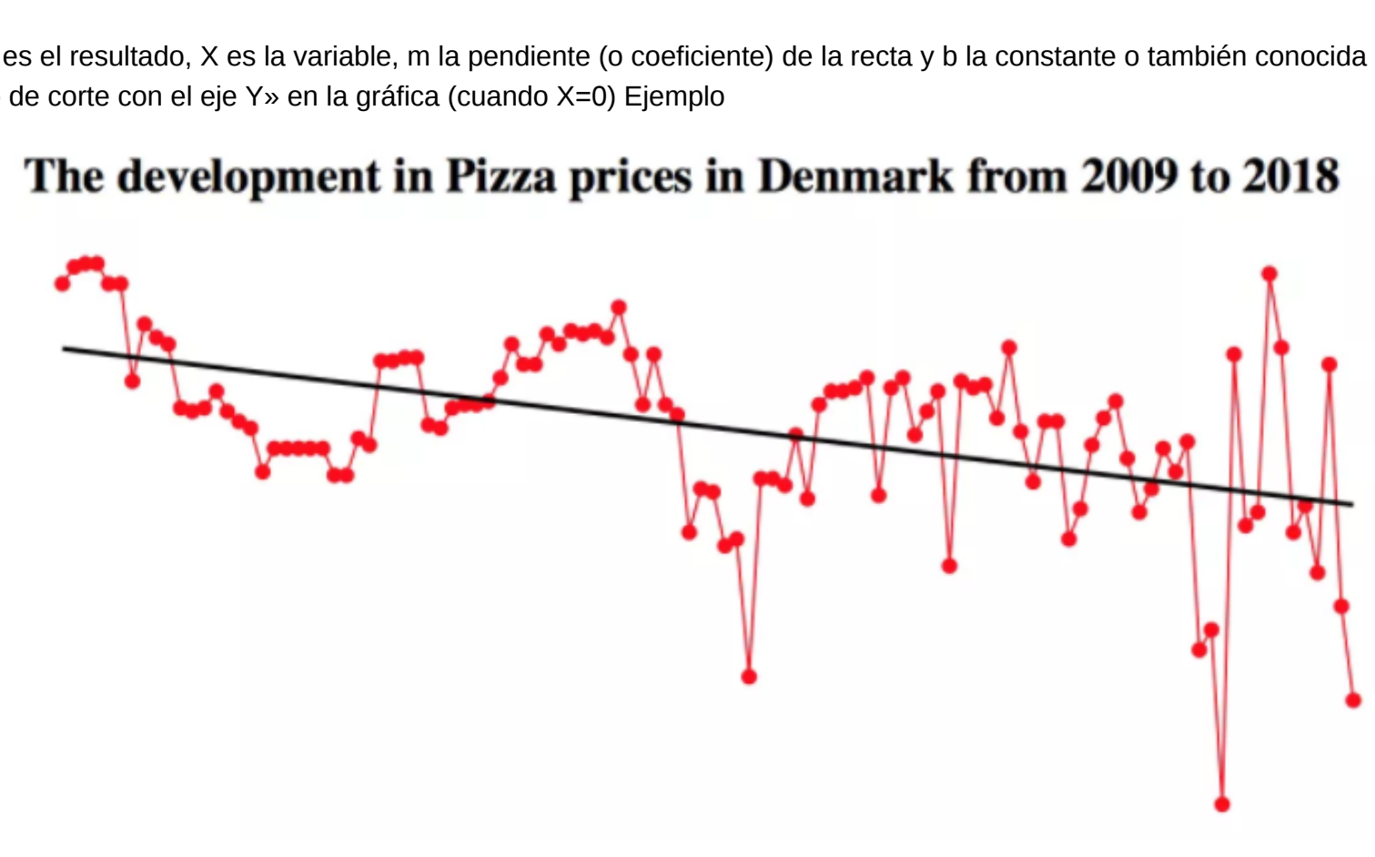
EL modelo lineal

La regresión lineal es un algoritmo de aprendizaje supervisado que se utiliza en Machine Learning y en estadística. En su versión más sencilla, lo que haremos es «dibujar una recta» que nos indicará la tendencia de un conjunto de datos continuos.

Recordemos rápidamente la fórmula de la recta:

Y = mX + b

Donde Y es el resultado, X es la variable, m la pendiente (o coeficiente) de la recta y b la constante o también conocida como el «punto de corte con el eje Y» en la gráfica (cuando X=0) Ejemplo



Recordemos que los algoritmos de Machine Learning Supervisados, aprenden por sí mismos y -en este caso- a obtener automáticamente esa «recta» que buscamos con la tendencia de predicción. Para hacerlo se mide el error con respecto a los

In [9]:
```
x = list(df.iloc [:, 0]) # Fecha
y = list(df.iloc [:, 1]) # Numero de casos
# Creamos el objeto de Regresión Lineal
regr = linear_model.LinearRegression()

# Entrenamos nuestro modelo
regr.fit(np.array(x).reshape(-1, 1) ,y)

# Veamos los coeficienetes obtenidos, En nuestro caso, serán la Tangente
print('Coefficients: \n', regr.coef_)
# Este es el valor donde corta el eje Y (en X=0)
print('Independent term: \n', regr.intercept_)
# Error Cuadrado Medio
```

```
Coefficients:
 [678.23434836]
Independent term:
 -50243.115476957784
```

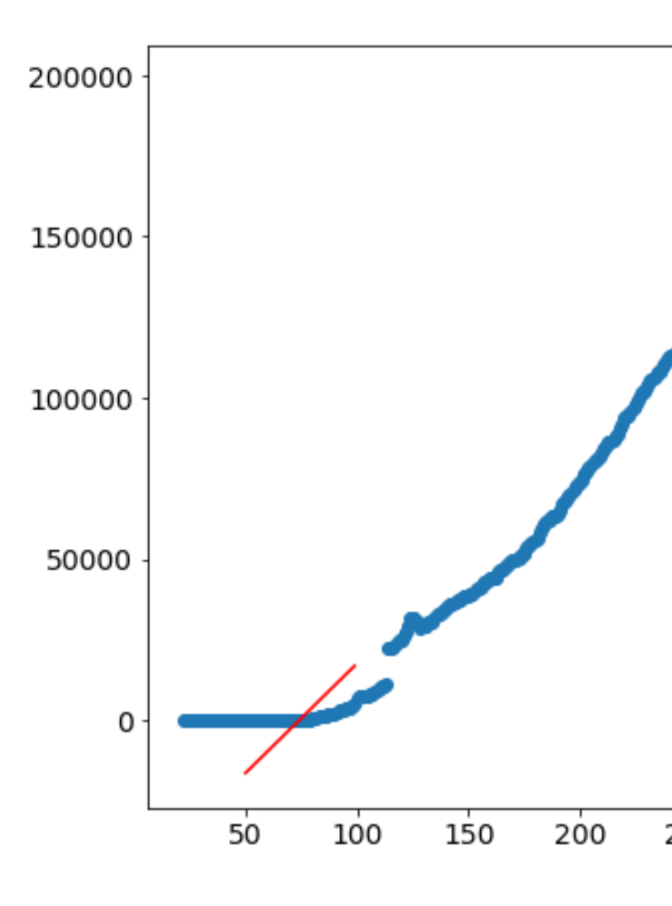
De la ecuación de la recta y = mX + b nuestra pendiente «m» es el coeficiente y el término independiente «b»

In [14]:
```
#Vamos a comprobar:
# Quiero predecir cuántos "Casos" voy a obtener por en el dia 100,
# según nuestro modelo, hacemos:
y_prediccion = regr.predict([[100]])
print('Para el dia 100 el numero de casos son:',int(y_prediccion))
```

```
Para el dia 100 el numero de casos son: 17580
```

In [20]:
```
#Graficar
plt.scatter(x, y)
x_real = np.array(range(50, 100))
print(x_real)
plt.plot(x_real, regr.predict(x_real.reshape(-1, 1)), color='red')
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]
```



El modelo logistico

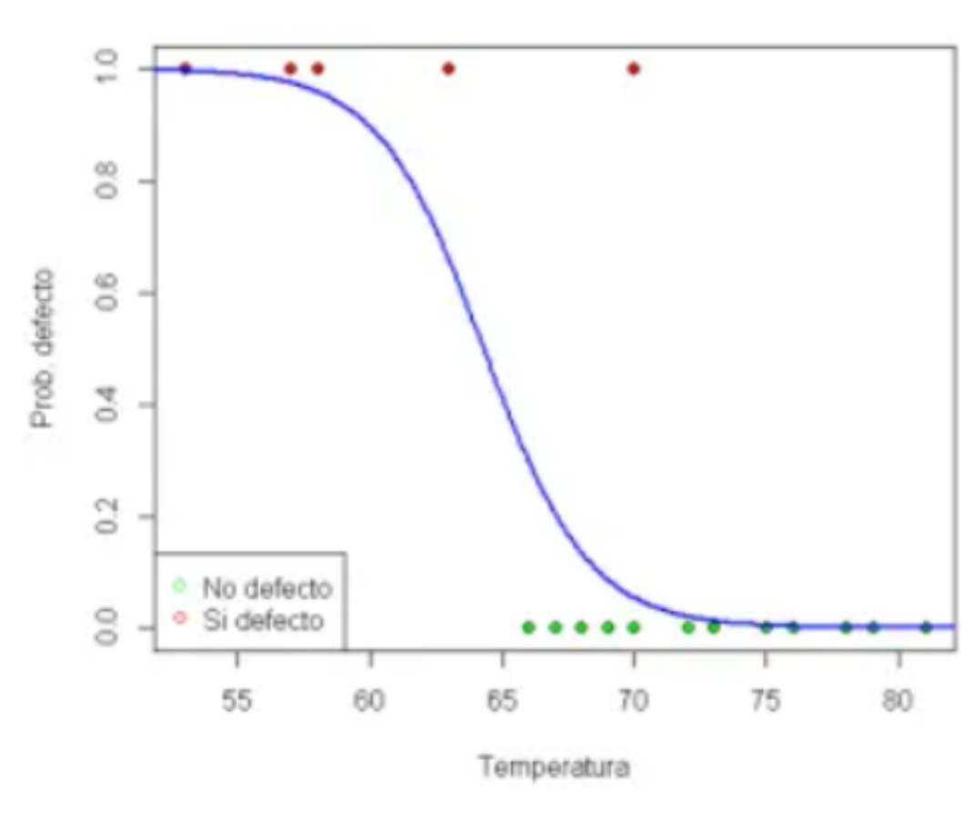
El modelo logísticoes una función matemática que aparece en diversos modelos de crecimiento de poblaciones, propagación de enfermedades epidémicas y difusión en redes sociales. La expresión más genérica de una función logística es:

$$f(x,a,b,c) = \frac{c}{1+e^{-(x-b)/a}}$$

En esta fórmula, tenemos la variable x que es el tiempo y tres parámetros: a, b, c .

- a se refiere a la velocidad de infección
- b es el día en que ocurrieron las infecciones máximas
- c es el número total de personas infectadas registradas al final de la infección

A continuacion se puede apreciar un ejemplo de regresion logistica

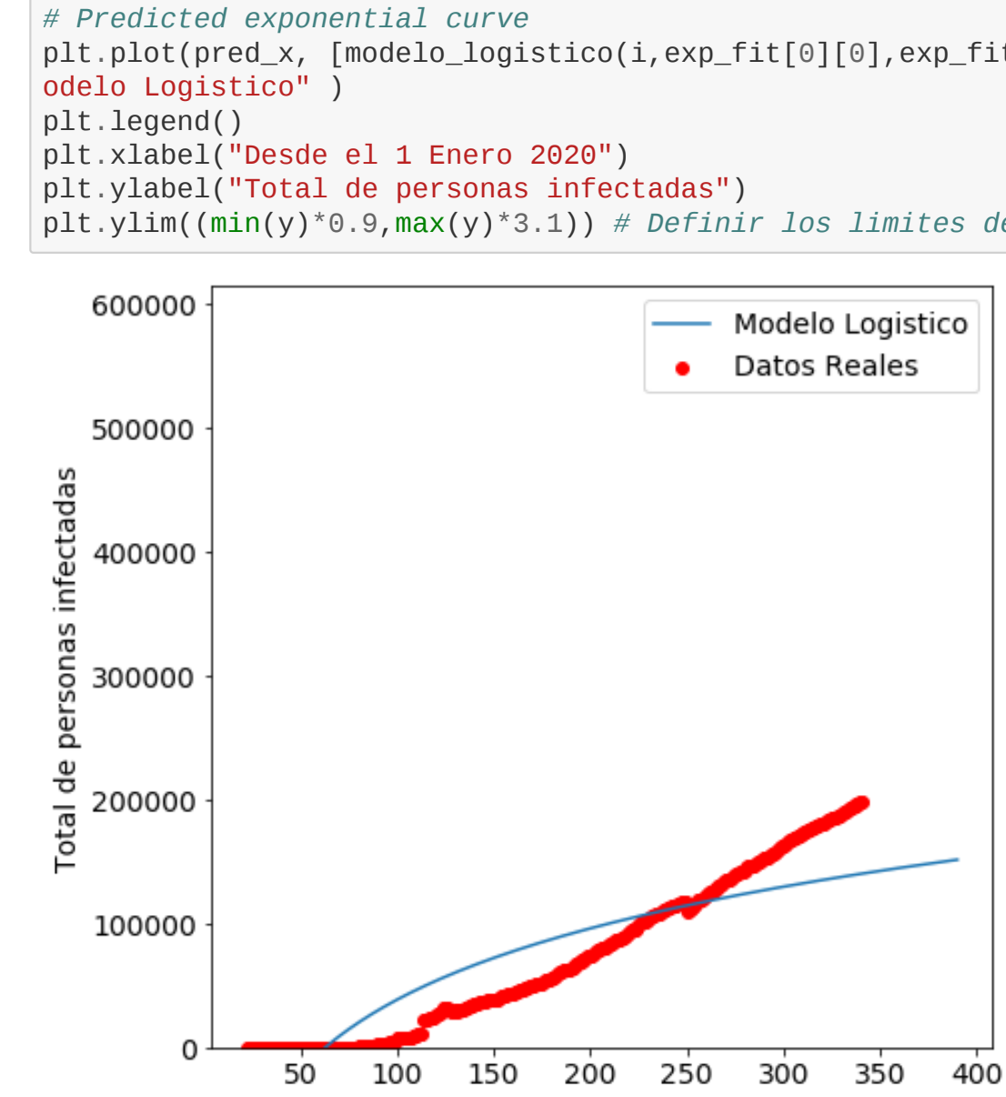


Definamos la función en Python y realicemos el procedimiento de ajuste de curva utilizado para el crecimiento logístico.

In [16]:
```
def modelo_logistico(x,a,b):
    return a+b*np.log(x)

exp_fit = curve_fit(modelo_logistico,x,y) #Extraemos los valores de los paramatros
print(exp_fit)
```

```
(array([-343832.62635712,   83019.89950144]), array([[ 1.73473472e+08, -3.39535647e+07],
       [-3.39535647e+07,  6.76480830e+06]]))
```

In [17]:
```
# Graficas
pred_x = list(range(min(x),max(x)+50)) # Predecir 50 dias mas
plt.rcParams['figure.figsize'] = [7, 7]
plt.rc('font', size=14)
# Real data
plt.scatter(x,y,label="Datos Reales",color="red")
# Predicted exponential curve
plt.plot(pred_x, [modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i in pred_x], label="M
odelo Logistico" )
plt.legend()
plt.xlabel("Desde el 1 Enero 2020")
plt.ylabel("Total de personas infectadas")
plt.ylim((min(y)*0.9,max(y)*3.1)) # Definir los limites de Y
```



Modelo exponencial

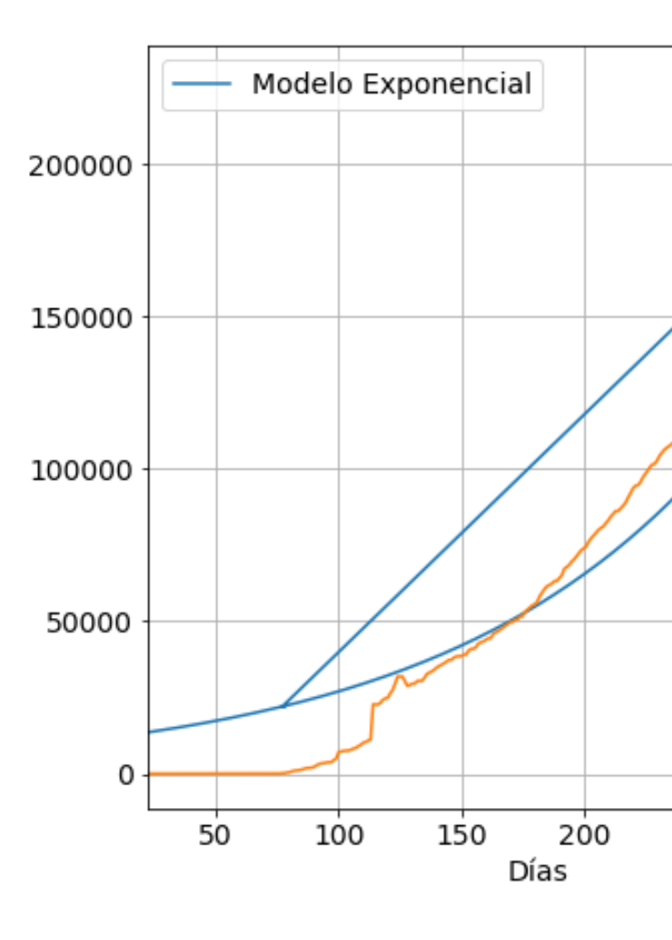
Un modelo, de fácil interpretación geométrica, que puede ser utilizado para la representación de los casos por cada día, es el modelo exponencial debido a que describe un crecimiento de infección que se detendrá en el futuro, el modelo exponencial describe un crecimiento de infección imparable . Por ejemplo, si un paciente infecta a 2 pacientes por día, después de 1 día tendremos 2 infecciones, 4 después de 2 días, 8 después de 3 y así sucesivamente.

$$N(t) = ae^{-b(t-c)^2}$$

En este modelo N(t) representa los casos obtenidos por días (casos confirmados o fallecimientos), t representa el tiempo trascurrido luego de haberse presentados los primeros casos y a, b y c son parámetros que pueden ser obtenidos (luego de realizar transformaciones matemáticas) aplicando el método de los MCL. En este modelo es de gran importancia el punto estacionario que muestra el cambio de comportamiento de la curva que pasa de un crecimiento a un decrecimiento.

In [41]:
```
# Implementar
# Función para añadir días a la predicción
def devolverDias(adicion):
    dias = x.copy()
    for dia in range(77,(77+adicion)):
        dias.append(dia)
    return dias

# Definición de función exponencial
def genExp(varIn, a, b):
    return a*np.exp(b*varIn)

exp_fit = curve_fit(genExp,x,y)
#print(exp_fit)

pred_x = list(range(0,max(x)+100))

# Creación de dataframes para el almacenamieto de los resultados
modeloExp = pd.DataFrame(columns=('Días', 'Modelo Exponencial'))
modeloPol = pd.DataFrame(columns=('Días', 'Modelo Polinomial'))
parametros, valoresCovarianza = curve_fit(genExp, x, y, p0=(0,0))

# Adicion de días para prediccion
dias = devolverDias(2)
puntos=[genExp(i,exp_fit[0][0],exp_fit[0][1]) for i in pred_x]
print ('Predicción a 7 días :', puntos[37], 'contagiados')
# Asignación de valores a la variable dependiente según la ecuación anterior
for pred in dias:
    modeloExp.loc[len(modeloExp)] = [pred,genExp(pred, parametros[0], parametros[1])]
# Gráfico del modelo exponencial
modeloExp.plot(x='Días', y='Modelo Exponencial')
plt.plot(df.loc[:,'date'],df.loc[:,'total_cases'])
plt.grid()
```

```
Predicción a 7 días : 20948.968214977245 contagiados
```



Modelo polinomial

Predicción de una variable de respuesta cuantitativa a partir de una variable predictora cuantitativa, donde la relación se modela como una función polinomial de orden n (esto significa que pueden tener de diferentes exponenciales o grados y se debe ir probando)
Se puede tener una ecuacion con diferentes grados

y = a0 + a1x + a2x^2 + a3x^3 + … + anx^n + ε

Ejemplo de una regresion polinomica de grado 4.

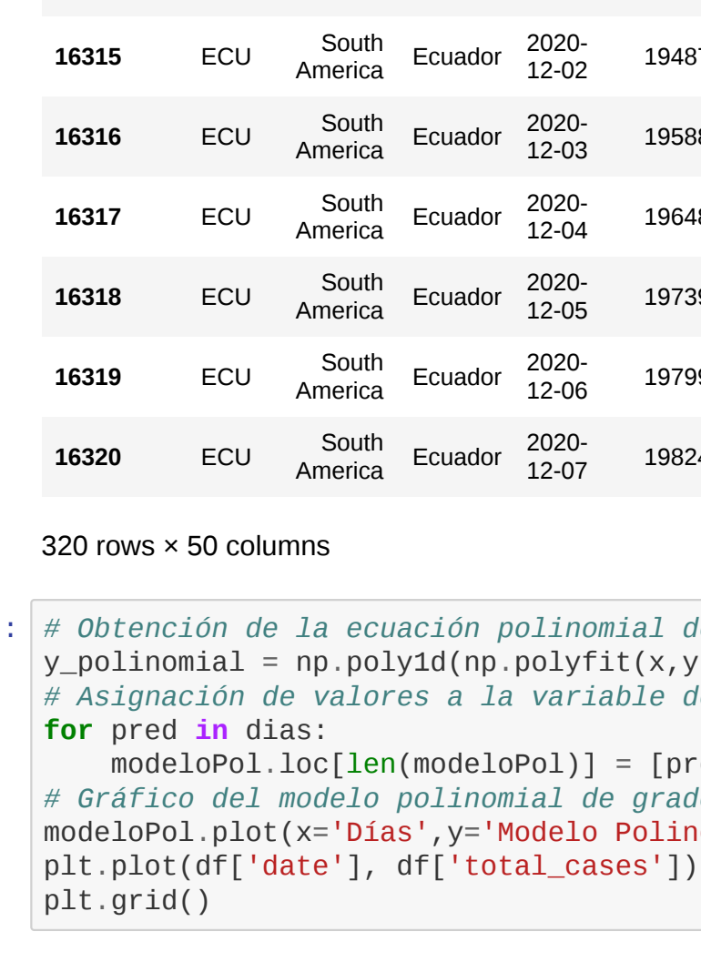
In [43]:
```
df = pd.read_csv('owid-covid-data.csv')
df = df[df['location'].isin(['Ecuador'])&(df['total_cases']!=0)] #Filtro la Informacion solo
para Ecuador
```

Out[43]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_de
16001	ECU	South America	Ecuador	2020-01-23	NaN	0.0	NaN	NaN	0.0	
16002	ECU	South America	Ecuador	2020-01-24	NaN	0.0	NaN	NaN	0.0	
16003	ECU	South America	Ecuador	2020-01-25	NaN	0.0	NaN	NaN	0.0	
16004	ECU	South America	Ecuador	2020-01-26	NaN	0.0	NaN	NaN	0.0	
16005	ECU	South America	Ecuador	2020-01-27	NaN	0.0	NaN	NaN	0.0	
16006	ECU	South America	Ecuador	2020-01-28	NaN	0.0	0.000	NaN	0.0	
16007	ECU	South America	Ecuador	2020-01-29	NaN	0.0	0.000	NaN	0.0	
16008	ECU	South America	Ecuador	2020-01-30	NaN	0.0	0.000	NaN	0.0	
16009	ECU	South America	Ecuador	2020-01-31	NaN	0.0	0.000	NaN	0.0	
16010	ECU	South America	Ecuador	2020-02-01	NaN	0.0	0.000	NaN	0.0	
16011	ECU	South America	Ecuador	2020-02-02	NaN	0.0	0.000	NaN	0.0	
16012	ECU	South America	Ecuador	2020-02-03	NaN	0.0	0.000	NaN	0.0	
16013	ECU	South America	Ecuador	2020-02-04	NaN	0.0	0.000	NaN	0.0	
16014	ECU	South America	Ecuador	2020-02-05	NaN	0.0	0.000	NaN	0.0	
16015	ECU	South America	Ecuador	2020-02-06	NaN	0.0	0.000	NaN	0.0	
16016	ECU	South America	Ecuador	2020-02-07	NaN	0.0	0.000	NaN	0.0	
16017	ECU	South America	Ecuador	2020-02-08	NaN	0.0	0.000	NaN	0.0	
16018	ECU	South America	Ecuador	2020-02-09	NaN	0.0	0.000	NaN	0.0	
16019	ECU	South America	Ecuador	2020-02-10	NaN	0.0	0.000	NaN	0.0	
16020	ECU	South America	Ecuador	2020-02-11	NaN	0.0	0.000	NaN	0.0	
16021	ECU	South America	Ecuador	2020-02-12	NaN	0.0	0.000	NaN	0.0	
16022	ECU	South America	Ecuador	2020-02-13	NaN	0.0	0.000	NaN	0.0	
16023	ECU	South America	Ecuador	2020-02-14	NaN	0.0	0.000	NaN	0.0	
16024	ECU	South America	Ecuador	2020-02-15	NaN	0.0	0.000	NaN	0.0	
16025	ECU	South America	Ecuador	2020-02-16	NaN	0.0	0.000	NaN	0.0	
16026	ECU	South America	Ecuador	2020-02-17	NaN	0.0	0.000	NaN	0.0	
16027	ECU	South America	Ecuador	2020-02-18	NaN	0.0	0.000	NaN	0.0	
16028	ECU	South America	Ecuador	2020-02-19	NaN	0.0	0.000	NaN	0.0	
16029	ECU	South America	Ecuador	2020-02-20	NaN	0.0	0.000	NaN	0.0	
16030	ECU	South America	Ecuador	2020-02-21	NaN	0.0	0.000	NaN	0.0	
...	...	...	...	...	...	...	...	...	...	
16291	ECU	South America	Ecuador	2020-11-08	174907.0	1421.0	816.143	12830.0	15.0	
16292	ECU	South America	Ecuador	2020-11-09	175269.0	362.0	815.286	12839.0	9.0	
16293	ECU	South America	Ecuador	2020-11-10	175711.0	442.0	800.143	12849.0	10.0	
16294	ECU	South America	Ecuador	2020-11-11	176630.0	919.0	742.429	12920.0	71.0	
16295	ECU	South America	Ecuador	2020-11-12	177513.0	883.0	818.571	12946.0	26.0	
16296	ECU	South America	Ecuador	2020-11-13	178674.0	1161.0	880.857	12977.0	31.0	
16297	ECU	South America	Ecuador	2020-11-14	179627.0	953.0	877.286	12997.0	20.0	
16298	ECU	South America	Ecuador	2020-11-15	180295.0	668.0	769.714	13008.0	11.0	
16299	ECU	South America	Ecuador	2020-11-16	180676.0	381.0	772.429	13016.0	8.0	
16300	ECU	South America	Ecuador	2020-11-17	181104.0	428.0	770.429	13025.0	9.0	
16301	ECU	South America	Ecuador	2020-11-18	182250.0	1146.0	802.857	13052.0	27.0	
16302	ECU	South America	Ecuador	2020-11-19	183246.0	996.0	819.000	13073.0	21.0	
16303	ECU	South America	Ecuador	2020-11-20	183840.0	594.0	738.000	13095.0	22.0	
16304	ECU	South America	Ecuador	2020-11-21	184876.0	1036.0	749.857	13139.0	44.0	
16305	ECU	South America	Ecuador	2020-11-22	185643.0	767.0	764.000	13201.0	62.0	
16306	ECU	South America	Ecuador	2020-11-23	185944.0	301.0	752.571	13225.0	24.0	
16307	ECU	South America	Ecuador	2020-11-24	186436.0	492.0	761.714	13264.0	39.0	
16308	ECU	South America	Ecuador	2020-11-25	187230.0	794.0	711.429	13288.0	24.0	
16309	ECU	South America	Ecuador	2020-11-26	188138.0	908.0	698.857	13316.0	28.0	
16310	ECU	South America	Ecuador	2020-11-27	189534.0	1396.0	813.429	13358.0	42.0	
16311	ECU	South America	Ecuador	2020-11-28	190909.0	1375.0	861.857	13371.0	13.0	
16312	ECU	South America	Ecuador	2020-11-29	192117.0	1208.0	924.857	13423.0	52.0	
16313	ECU	South America	Ecuador	2020-11-30	192685.0	568.0	963.000	13461.0	38.0	
16314	ECU	South America	Ecuador	2020-12-01	193673.0	988.0	1033.857	13501.0	40.0	
16315	ECU	South America	Ecuador	2020-12-02	194876.0	1203.0	1092.286	13562.0	61.0	
16316	ECU	South America	Ecuador	2020-12-03	195884.0	1008.0	1106.571	13612.0	50.0	
16317	ECU	South America	Ecuador	2020-12-04	196482.0	598.0	992.571	13696.0	84.0	
16318	ECU	South America	Ecuador	2020-12-05	197391.0	909.0	926.000	13756.0	60.0	
16319	ECU	South America	Ecuador	2020-12-06	197998.0	607.0	840.143	13778.0	22.0	
16320	ECU	South America	Ecuador	2020-12-07	198244.0	246.0	794.143	13780.0	2.0	

320 rows × 50 columns

In [45]:
```
# Obtención de la ecuación polinomial de grado 'n'
y_polinomial = np.poly1d(np.polyfit(x,y,6))
# Asignación de valores a la variable dependiente según la ecuación anterior
for pred in dias:
    modeloPol.loc[len(modeloPol)] = [pred, y_polinomial(pred)]
# Gráfico del modelo polinomial de grado 9
modeloPol.plot(x='Días',y='Modelo Polinomial')
plt.plot(df['date'], df['total_cases'])
plt.grid()
```



Analisis

Con la infomacion que encontramos en internet, y con la ayudar de los modelos matematicos podemos predecir el numero de personas que has sido infectadas durante la crisis, y para simular o predecir la evolución de los casos en el futuro y poder tomar medidas.

Conclusiones

El empleo de modelos matematicos para las enfermedads, va creciendo en un grado significativo, debido a que nos proporciona informacion util para tomar decisiones, para poder predecir la cantidad de personas infectadas o fallecidas y buscar distinas maneras de convatir la enfermedas y proteger a la sociedad.Los modelos matematicos no solo se usan en los casos de enfermedas sino tambien se pueden usar para buscar predicir los casos economicos que atraviesa el país , etc.

Criterio personal (politico, economico y social de la situacion)

Dentro del ambito economico, las personas han experimentado una disminucion miticulosa , ya que ha abarcado a millones de personas a la pobreza, debia que la magnitud de perdidas dentro de la economia del país, una vez que tomamos en cuenta la situacion que el país esta atrevesando debemos de buscar la forma de superar la curva de aprendizaje, y buscar los beneficios que representa la visualicion de prediciones, basando en los hechos dentro de la sociedad.

Referencias

- https://www.researchgate.net/publication/340092755_Infeccion_del_Covid-19_en_Colombia_Una_comparacion_de_modelos_logisticos_y_exponenciales_aplicados_a_la_infeccion_por_el_virus_en
- https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/