Universida Politecnica Salesiana

Nombre: Erika Morocho

Asignatura: Simulacion

Fecha: 23/11/2020

In [34]:

```
import math
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

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
from scipy.optimize import curve_fit
from scipy.integrate import solve_ivp
from scipy.optimize import minimize
from scipy.integrate import odeint

%matplotlib inline
```

In [4]:

```
df = pd.read_csv('owid-covid-data.csv').fillna(0)
ndf= df.loc[(df['location'] == 'Costa Rica') & (df['total_cases'] != 0)]
ndf
```

Out[4]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_de
12074	CRI	North America	Costa Rica	2020-03-07	1.0	1.0	0.000	0.0	0.0	
12075	CRI	North America	Costa Rica	2020-03-09	5.0	4.0	0.000	0.0	0.0	
12076	CRI	North America	Costa Rica	2020-03-10	9.0	4.0	0.000	0.0	0.0	
12077	CRI	North America	Costa Rica	2020-03-11	13.0	4.0	0.000	0.0	0.0	
12078	CRI	North America	Costa Rica	2020-03-12	22.0	9.0	0.000	0.0	0.0	
12079	CRI	North America	Costa Rica	2020-03-13	23.0	1.0	3.286	0.0	0.0	
12080	CRI	North America	Costa Rica	2020-03-14	26.0	3.0	3.571	0.0	0.0	
12081	CRI	North America	Costa Rica	2020-03-15	27.0	1.0	3.714	0.0	0.0	
12082	CRI	North America	Costa Rica	2020-03-16	35.0	8.0	4.286	0.0	0.0	
12083	CRI	North	Costa	2020-	41.0	6.0	4.571	0.0	0.0	

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_de
12083	CRI	America	Rica	03-17	41.0	6.0	4.571	0.0	0.0	
12084	CRI	North America	Costa Rica	2020-03-18	50.0	9.0	5.286	0.0	0.0	
12085	CRI	North America	Costa Rica	2020-03-19	69.0	19.0	6.714	1.0	1.0	
12086	CRI	North America	Costa Rica	2020-03-20	87.0	18.0	9.143	1.0	0.0	
12087	CRI	North America	Costa Rica	2020-03-21	113.0	26.0	12.429	2.0	1.0	
12088	CRI	North America	Costa Rica	2020-03-22	117.0	4.0	12.857	2.0	0.0	
12089	CRI	North America	Costa Rica	2020-03-23	134.0	17.0	14.143	2.0	0.0	
12090	CRI	North America	Costa Rica	2020-03-24	158.0	24.0	16.714	2.0	0.0	
12091	CRI	North America	Costa Rica	2020-03-25	177.0	19.0	18.143	2.0	0.0	
12092	CRI	North America	Costa Rica	2020-03-26	201.0	24.0	18.857	2.0	0.0	
12093	CRI	North America	Costa Rica	2020-03-27	231.0	30.0	20.571	2.0	0.0	
12094	CRI	North America	Costa Rica	2020-03-28	263.0	32.0	21.429	2.0	0.0	
12095	CRI	North America	Costa Rica	2020-03-29	295.0	32.0	25.429	2.0	0.0	
12096	CRI	North America	Costa Rica	2020-03-30	314.0	19.0	25.714	2.0	0.0	
12097	CRI	North America	Costa Rica	2020-03-31	330.0	16.0	24.571	2.0	0.0	
12098	CRI	North America	Costa Rica	2020-04-01	347.0	17.0	24.286	2.0	0.0	
12099	CRI	North America	Costa Rica	2020-04-02	375.0	28.0	24.857	2.0	0.0	
12100	CRI	North America	Costa Rica	2020-04-03	396.0	21.0	23.571	2.0	0.0	
12101	CRI	North America	Costa Rica	2020-04-04	416.0	20.0	21.857	2.0	0.0	
12102	CRI	North America	Costa Rica	2020-04-05	435.0	19.0	20.000	2.0	0.0	
12103	CRI	North America	Costa Rica	2020-04-06	454.0	19.0	20.000	2.0	0.0	
...	...	...	...	...	...	...	...	...	...	
12305	CRI	North America	Costa Rica	2020-10-25	103088.0	1262.0	1082.000	1282.0	17.0	
12306	CRI	North America	Costa Rica	2020-10-26	103088.0	0.0	1082.000	1282.0	0.0	
12307	CRI	North America	Costa Rica	2020-10-27	104460.0	1372.0	1055.000	1312.0	30.0	
12308	CRI	North America	Costa Rica	2020-10-28	105322.0	862.0	1057.143	1329.0	17.0	
12309	CRI	North America	Costa Rica	2020-10-29	106553.0	1231.0	1018.286	1340.0	11.0	
12310	CRI	North America	Costa Rica	2020-10-30	107570.0	1017.0	993.429	1357.0	17.0	
12311	CRI	North America	Costa Rica	2020-10-31	108866.0	1296.0	1005.714	1371.0	14.0	
		North	Costa	2020-						

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_de
12312	CRI	North America	Costa Rica	2020-11-01	109971.0	1105.0	983.286	1385.0	14.0	
12313	CRI	North America	Costa Rica	2020-11-02	109971.0	0.0	983.286	1385.0	0.0	
12314	CRI	North America	Costa Rica	2020-11-03	111257.0	1286.0	971.000	1404.0	19.0	
12315	CRI	North America	Costa Rica	2020-11-04	112120.0	863.0	971.143	1419.0	15.0	
12316	CRI	North America	Costa Rica	2020-11-05	113261.0	1141.0	958.286	1431.0	12.0	
12317	CRI	North America	Costa Rica	2020-11-06	114367.0	1106.0	971.000	1444.0	13.0	
12318	CRI	North America	Costa Rica	2020-11-07	115417.0	1050.0	935.857	1453.0	9.0	
12319	CRI	North America	Costa Rica	2020-11-08	116363.0	946.0	913.143	1464.0	11.0	
12320	CRI	North America	Costa Rica	2020-11-09	116363.0	0.0	913.143	1464.0	0.0	
12321	CRI	North America	Costa Rica	2020-11-10	117587.0	1224.0	904.286	1491.0	27.0	
12322	CRI	North America	Costa Rica	2020-11-11	118566.0	979.0	920.857	1502.0	11.0	
12323	CRI	North America	Costa Rica	2020-11-12	119768.0	1202.0	929.571	1513.0	11.0	
12324	CRI	North America	Costa Rica	2020-11-13	120939.0	1171.0	938.857	1527.0	14.0	
12325	CRI	North America	Costa Rica	2020-11-14	122123.0	1184.0	958.000	1537.0	10.0	
12326	CRI	North America	Costa Rica	2020-11-15	123223.0	1100.0	980.000	1546.0	9.0	
12327	CRI	North America	Costa Rica	2020-11-16	123223.0	0.0	980.000	1546.0	0.0	
12328	CRI	North America	Costa Rica	2020-11-17	124592.0	1369.0	1000.714	1566.0	20.0	
12329	CRI	North America	Costa Rica	2020-11-18	125590.0	998.0	1003.429	1578.0	12.0	
12330	CRI	North America	Costa Rica	2020-11-19	127012.0	1422.0	1034.857	1588.0	10.0	
12331	CRI	North America	Costa Rica	2020-11-20	128231.0	1219.0	1041.714	1599.0	11.0	
12332	CRI	North America	Costa Rica	2020-11-21	129418.0	1187.0	1042.143	1608.0	9.0	
12333	CRI	North America	Costa Rica	2020-11-22	129418.0	0.0	885.000	1608.0	0.0	
12334	CRI	North America	Costa Rica	2020-11-23	129418.0	0.0	885.000	1608.0	0.0	

261 rows × 50 columns

In [5]:

```
df = df[df['location'].isin(['Costa Rica'])] #Filtro la Informacion solo para Ecuador
df = df.loc[:,['date','total_cases','iso_code']] #Selecciono las columnas de analasis
FMT = '%Y-%m-%d'
date = df['date']
df['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("2019-12
-30", FMT)).days)

df
```

Out[5]:

	date	total_cases	iso_code
12074	68	1.0	CRI
12075	70	5.0	CRI
12076	71	9.0	CRI
12077	72	13.0	CRI
12078	73	22.0	CRI
12079	74	23.0	CRI
12080	75	26.0	CRI
12081	76	27.0	CRI
12082	77	35.0	CRI
12083	78	41.0	CRI
12084	79	50.0	CRI
12085	80	69.0	CRI
12086	81	87.0	CRI
12087	82	113.0	CRI
12088	83	117.0	CRI
12089	84	134.0	CRI
12090	85	158.0	CRI
12091	86	177.0	CRI
12092	87	201.0	CRI
12093	88	231.0	CRI
12094	89	263.0	CRI
12095	90	295.0	CRI
12096	91	314.0	CRI
12097	92	330.0	CRI
12098	93	347.0	CRI
12099	94	375.0	CRI
12100	95	396.0	CRI
12101	96	416.0	CRI
12102	97	435.0	CRI
12103	98	454.0	CRI
...	...	...	...
12305	300	103088.0	CRI
12306	301	103088.0	CRI
12307	302	104460.0	CRI
12308	303	105322.0	CRI
12309	304	106553.0	CRI
12310	305	107570.0	CRI
12311	306	108866.0	CRI
12312	307	109971.0	CRI
12313	308	109971.0	CRI
12314	309	111257.0	CRI
12315	310	112120.0	CRI

	date	total_cases	iso_code
12316	311	113261.0	CRI
12317	312	114367.0	CRI
12318	313	115417.0	CRI
12319	314	116363.0	CRI
12320	315	116363.0	CRI
12321	316	117587.0	CRI
12322	317	118566.0	CRI
12323	318	119768.0	CRI
12324	319	120939.0	CRI
12325	320	122123.0	CRI
12326	321	123223.0	CRI
12327	322	123223.0	CRI
12328	323	124592.0	CRI
12329	324	125590.0	CRI
12330	325	127012.0	CRI
12331	326	128231.0	CRI
12332	327	129418.0	CRI
12333	328	129418.0	CRI
12334	329	129418.0	CRI

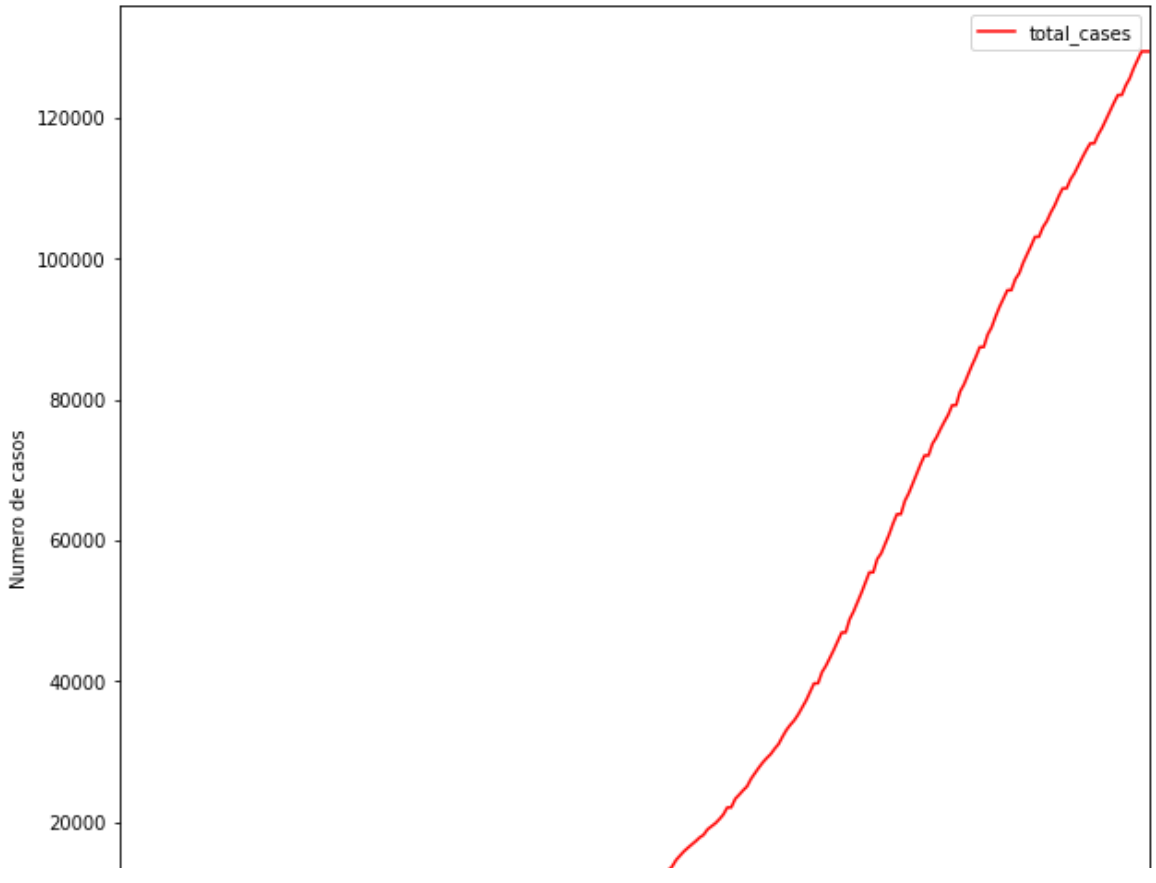
261 rows × 3 columns

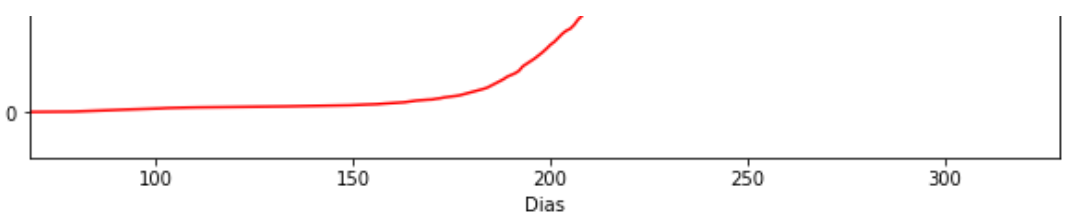
In [6]:

```
plt.rcParams['figure.figsize']=[10,10]
ax=df.plot(x='date',y='total_cases',color='red')
ax.set_xlabel("Dias")
ax.set_ylabel("Numero de casos")
```

Out[6]:

```
Text(0, 0.5, 'Numero de casos')
```





Regresion Lineal

In [7]:

```
x = list(df.iloc [:, 0]) # Fecha
y = list(df.iloc [:, 1]) # Numero de casos

# Creamos el objeto de Regresión Lineal
modelo = linear_model.LinearRegression()

# Entrenamos nuestro modelo
modelo.fit(np.array(x).reshape(-1, 1),y)
# Veamos los coeficienetes obtenidos, En nuestro caso, serán la Tangente
print('Coefficients: \n', modelo.coef_)
# Este es el valor donde corta el eje Y (en X=0)
print('Independent term: \n', modelo.intercept_)
```

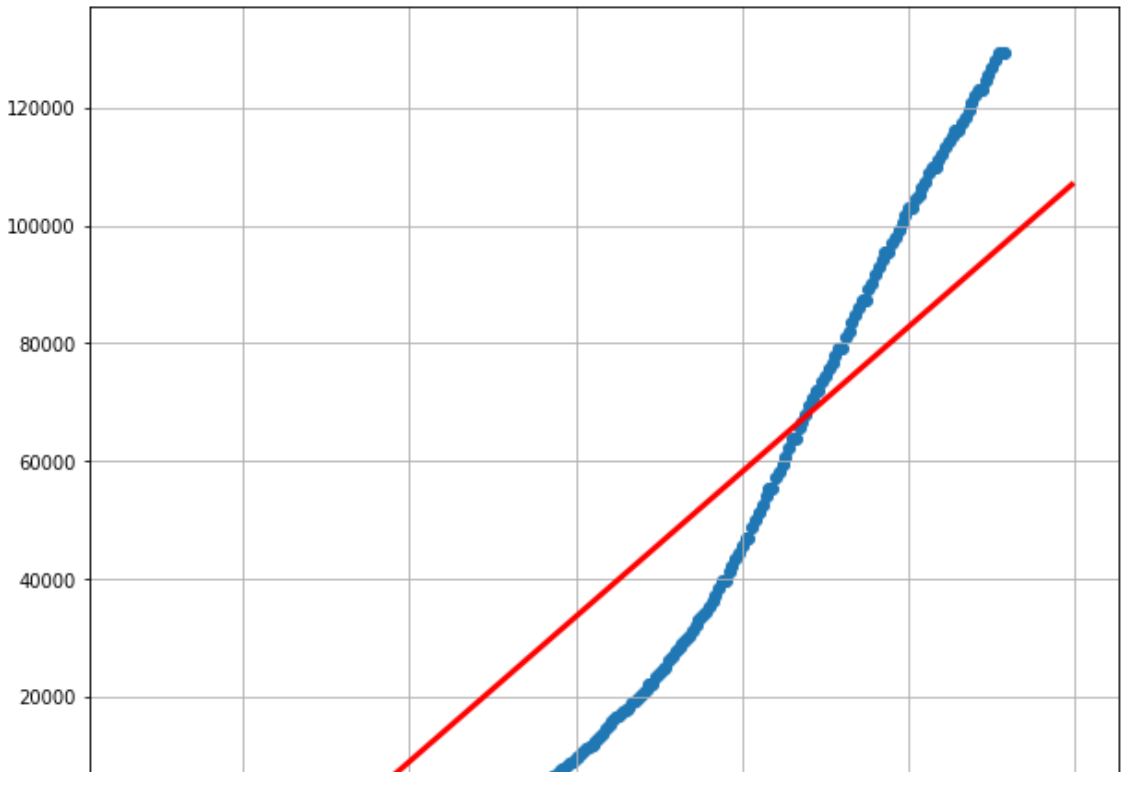
```
Coefficients:
 [492.66905898]
Independent term:
 -64990.0443879466
```

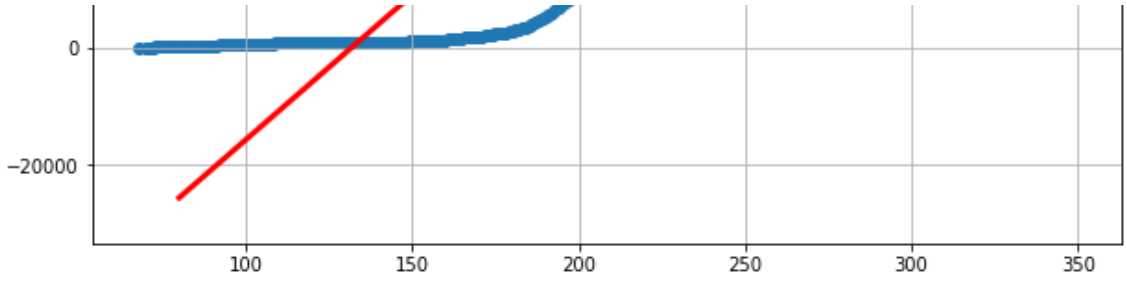
In [8]:

```
y_prediccion = modelo.predict([[len(x)+7]])
```

In [9]:

```
#Graficar
plt.scatter(x, y)
x_real = np.array(range(80, 350))
plt.plot(x_real, modelo.predict(x_real.reshape(-1, 1)), color='red',linewidth=3.0)
plt.grid()
plt.show()
print("Prediccion Proxima Semana: ")
print(int(y_prediccion))
```






```
Prediccion Proxima Semana:
67045
```

Regresion Exponencial

In [10]:

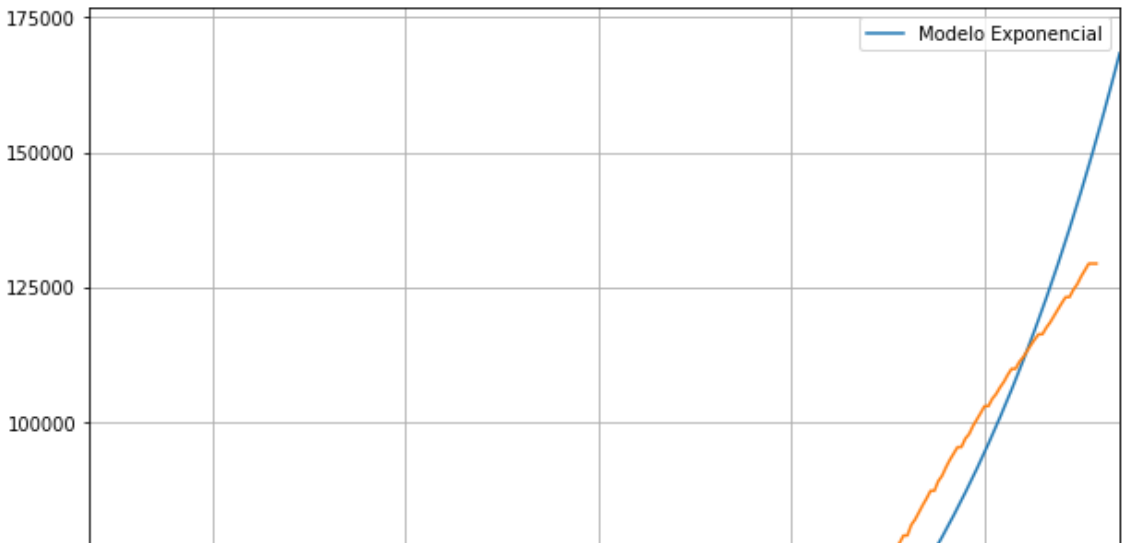
```
c=-2

def exponencial(x, a, b):
    return a * np . exp ( - b * (x  + c))

exp_fit = np.polyfit((x),np.log(y), deg=1)
print(exp_fit)
```

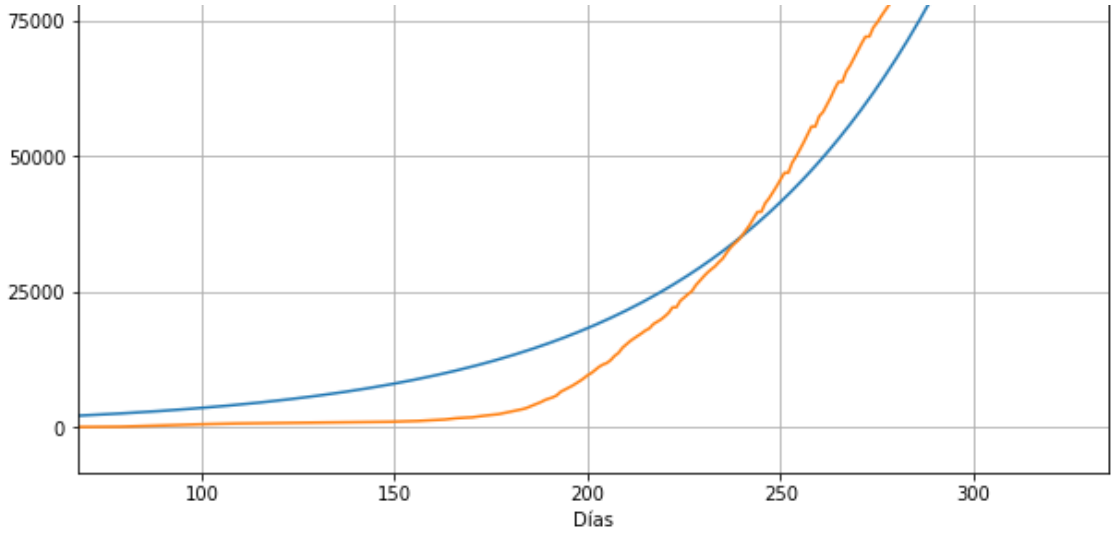
```
[0.03156511 2.41444012]
```

In [11]:

```
# Generar el modelo exponencial para la predicción de 7 días (día 75+45 = 120)
x_Exp = x[:]
y_Exp = y[:]
def genExp(varIn, a, b):
    return a*np.exp(b*varIn)
# Creación de dataframes para el almacenamieto de los resultados
modeloExp = pd.DataFrame(columns=('Días', 'Modelo Exponencial'))

parametros, valoresCovarianza = curve_fit(genExp, x_Exp, y_Exp, p0=(0,0))

# Asignación de valores a la variable dependiente según la ecuación anterior
pred_x = np.array(list(range(min(x),max(x)+7)))

for pred in pred_x:
    modeloExp.loc[len(modeloExp)] = [pred,genExp(pred, parametros[0], parametros[1])]
# Gráfico del modelo exponencial
modeloExp.plot(x='Días', y='Modelo Exponencial')
plt.plot(x_Exp,y_Exp)
plt.grid()

print('Prediccion',y[len(y)-1])
```

```
Prediccion 129418.0
```





In [12]:

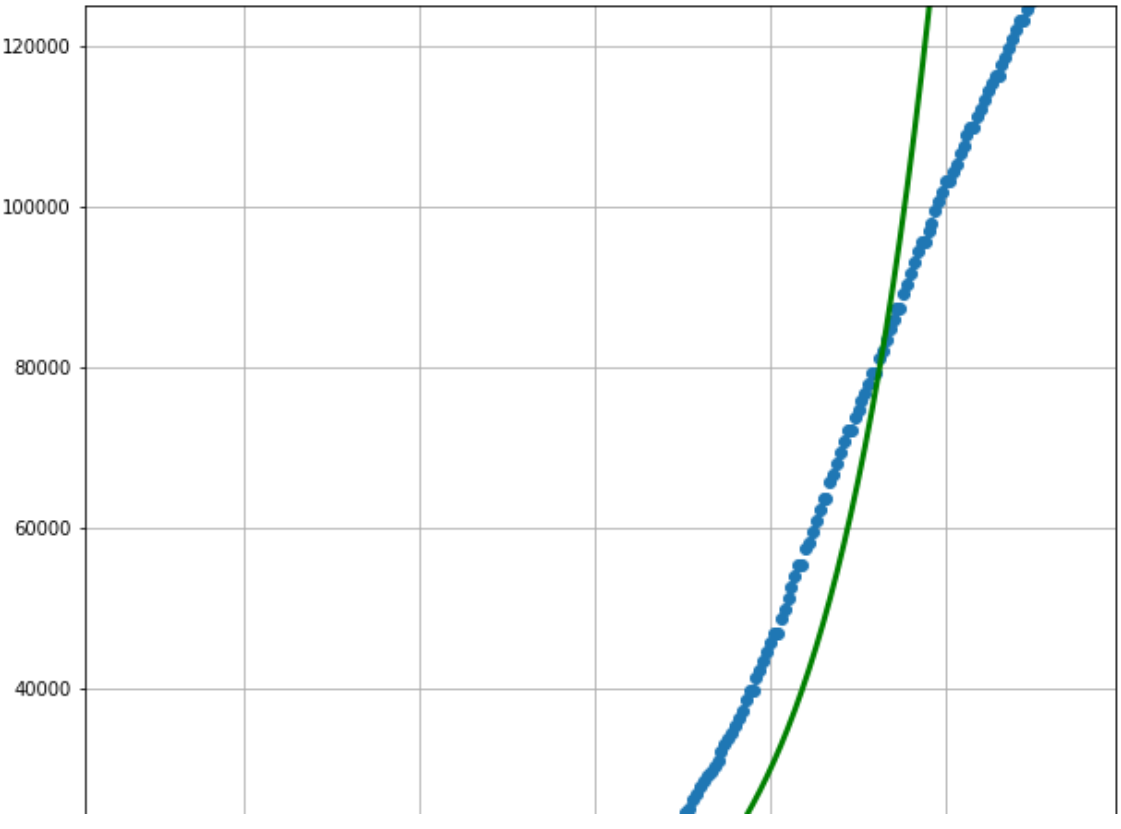
```
c=-2

def exponencial(x, a, b):
    return a * np . exp ( - b * (x  + c))

exp_fit = np.polyfit((x),np.log(y), deg=1)
print(exp_fit)
```

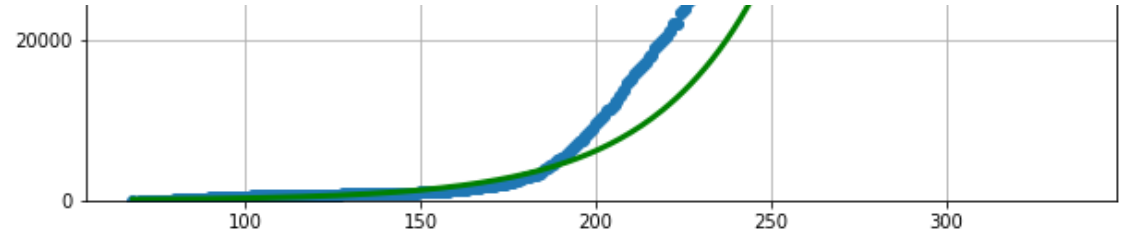
```
[0.03156511 2.41444012]
```

In [13]:

```
pred_x = np.array(list(range(min(x),max(x)+7)))
yx = np.exp(exp_fit[1]) * np.exp(exp_fit[0]*pred_x)

plt.plot(x,y,'o')
plt.plot(pred_x,yx,color="green",linewidth=3.0)
plt.ylim(-10,125000)

plt.grid(True)

print('Prediccion',yx[len(yx)-1])
```

```
Prediccion 437465.02531701466
```



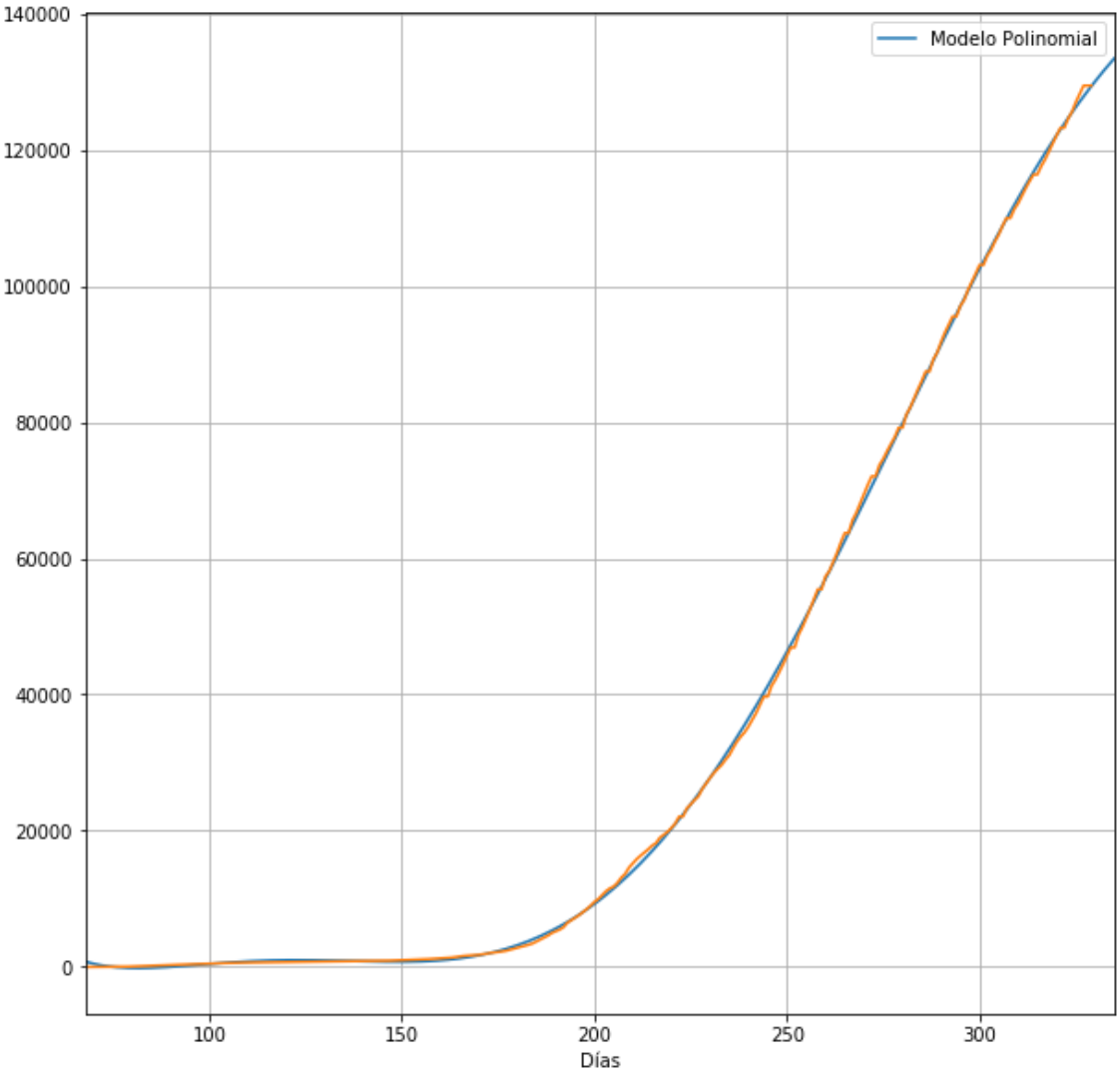


Regresion Polinomica

In [14]:

```
# Generar el modelo polinómico para la predicción de 7 días (día 75+45 = 120)
modeloPol = pd.DataFrame(columns=('Días', 'Modelo Polinomial'))
# 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 pred_x:
    modeloPol.loc[len(modeloPol)] = [pred, y_polinomial(pred)]
# Gráfico del modelo polinomial de grado 9
modeloPol.plot(x='Días',y='Modelo Polinomial')
plt.plot(x, y)
plt.grid()
print("PREDICCION: " , round(y_polinomial(len(x)+7),5))
```

```
PREDICCION:  65973.02008
```



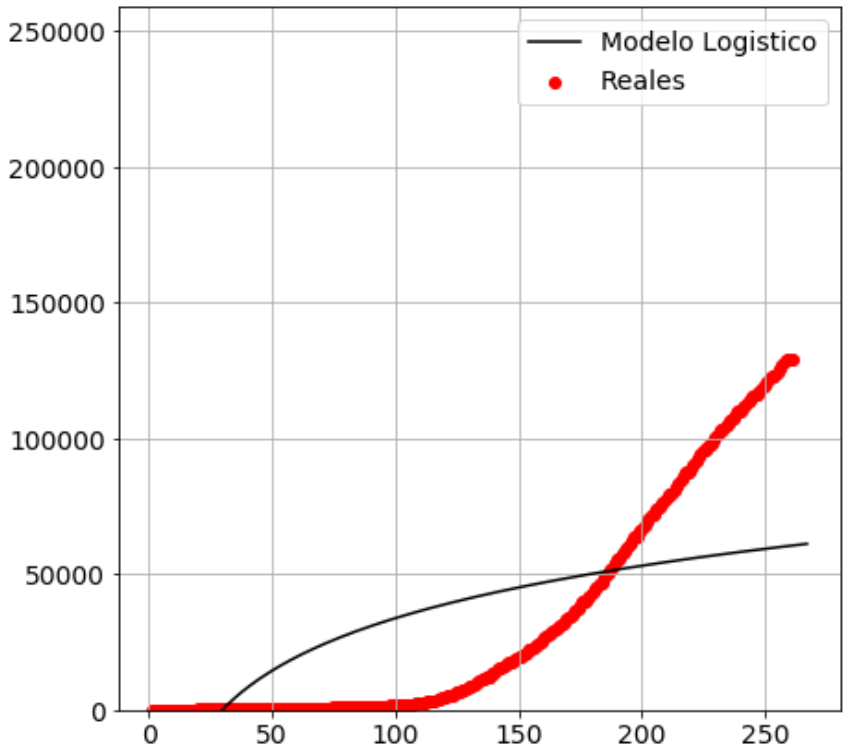
Regresion Logistica

In [79]:

```
from scipy.optimize import curve_fit
```

```
from sklearn.linear_model import LogisticRegression


def modelo_logistico(x,a,b):
    return a+b*np.log(x)

x=np.arange(1,len(df)+1,1)
y=np.array(df.values[:,1])

exp = curve_fit(modelo_logistico,x,y)

pred_x = list(range(min(x),max(x)+7))
plt.rcParams['figure.figsize'] = [7, 7]
plt.rc('font', size=14)

plt.scatter(x,y,label="Reales",color="red")

val = [modelo_logistico(i,exp[0][0],exp[0][1]) for i in pred_x]
plt.plot(pred_x, [modelo_logistico(i,exp[0][0],exp[0][1]) for i in pred_x], label="Model
o Logistico",color="black")
plt.legend()
print(val[len(pred_x)-1])

plt.ylim(0,max(y)*2)
plt.grid()
plt.show()
```

```
61151.203284706295
```



Probabilistico

In [75]:

```
df = pd.read_csv('owid-covid-data.csv').fillna(0)
df = df[df['location'].isin(['Costa Rica'])]
df = df.loc[:,['date','total_cases']]
FMT = '%Y-%m-%d'
date = df['date']
df['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("2020-01
-01",FMT)).days)
df
```

Out[75]:

	date	total_cases

	date	total_cases
12074	66	1.0
12075	68	5.0
12076	69	9.0
12077	70	13.0
12078	71	22.0
12079	72	23.0
12080	73	26.0
12081	74	27.0
12082	75	35.0
12083	76	41.0
12084	77	50.0
12085	78	69.0
12086	79	87.0
12087	80	113.0
12088	81	117.0
12089	82	134.0
12090	83	158.0
12091	84	177.0
12092	85	201.0
12093	86	231.0
12094	87	263.0
12095	88	295.0
12096	89	314.0
12097	90	330.0
12098	91	347.0
12099	92	375.0
12100	93	396.0
12101	94	416.0
12102	95	435.0
12103	96	454.0
...	...	...
12305	298	103088.0
12306	299	103088.0
12307	300	104460.0
12308	301	105322.0
12309	302	106553.0
12310	303	107570.0
12311	304	108866.0
12312	305	109971.0
12313	306	109971.0
12314	307	111257.0
12315	308	112120.0
12316	309	113261.0
12317	310	114367.0
12318	311	115417.0

	date	total_cases
12319	312	116363.0
12320	313	116363.0
12321	314	117587.0
12322	315	118566.0
12323	316	119768.0
12324	317	120939.0
12325	318	122123.0
12326	319	123223.0
12327	320	123223.0
12328	321	124592.0
12329	322	125590.0
12330	323	127012.0
12331	324	128231.0
12332	325	129418.0
12333	326	129418.0
12334	327	129418.0

261 rows × 2 columns

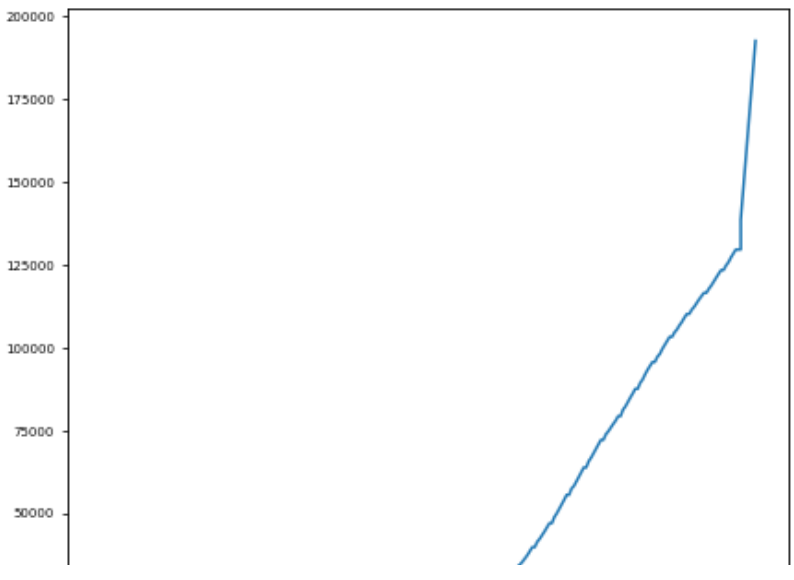
In [76]:

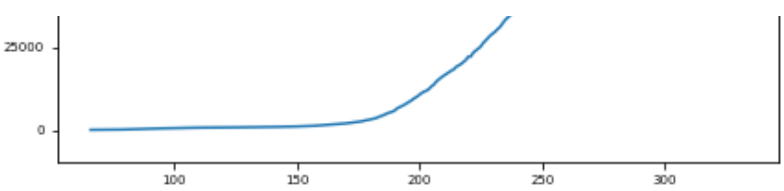
```
filtro = df['total_cases']
media = filtro.mean()
mediana = filtro.median()
print('medina:',mediana)
print('media:',media)
```

```
medina: 8986.0
media: 33049.21072796935
```

In [21]:

```
y = list(df.iloc [:, 1]) # Total casos
x = list(df.iloc [:, 0]) # Dias
for i in range(x[-1], x[-1]+7):
    x.append(i)
    y.append(int(y[-1] + mediana))
    prediccion_siguiente = int(y[-1] + mediana)
plt.plot(x, y)
plt.show()

print('Predicion Total de los casos ',prediccion_siguiente)
```






```
Predicion Total de los casos  201306
```

Cual tine una mejor prediccion

El modelo que tiene mayor prediccion es el polinomial, dando que se adapto mejor a la curva,brindo un valor exacto de contagiados dentro de el lapso de los 7 dias, ya que la grafia va creciendo y se va ajustando a los presicion de los los datos.

Ventajas y desventajas de los modelos

Ventajas

Mod.Lineal

- **una de las ventajas que tinee el modelo lineal, es que nos permite hacer una prediccion del comportamiento desde un punto determinado,muy sencillo de entender y rapido de modelar.**

Mod.Exponencial

- **Es un proceso que se ajusta mejor al conjunto de datos, ya que describe el crecimeinto de infeccion imparable.**

Mod.Polinomial

- **Funciona con cualquier tamaño de muestra.**
- **Trabaja bien sobre datos no lineales.**

Mod.Logaritmica

- **Es utilizado para describir el crecimiento de la poblacion.**
- **Es un modelo facil de clasificar**

Desventajas

Mod.Lineal

- **una de las desventajas que tiene es que no se puede modelar relaciones complejas no lineales**

Mod.Exponencial

- **En el modelo exponecial los valores de grado depende de la presicion, ya sea en el creciento.**

Mod.Polinomial

- **Se requiere elegir el grado correcto del polinomio para una buena relación sesgo/varianza.**

Mod.Logaritmica

- **Este modelo no resuelve proeblemas no lineales, por lo que su presicion no es buena.**

Principal problema del modelo probabilistico

Trabajar con el modelo probabilisticos, es adecuado pero cuando se tiene valores pequeños, pero al tener una gran cantidad de valores el modelo no predice de manera correcta, por lo que abordaria un marge de error en la prediccion de los datos.