Modelo probabilistico vs matematico

In [38]:
```
import pandas as pd
import numpy as np
from scipy.integrate import solve_ivp
from scipy.optimize import minimize
from scipy.integrate import odeint
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 [39]:
```
url = 'https://covid.ourworldindata.org/data/ecdc/new_cases.csv'
df = pd.read_csv(url)
df
```

24	2020-01-24	266	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN	...
25	2020-01-25	453	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN	...
26	2020-01-26	673	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN	...
27	2020-01-27	797	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN	...
28	2020-01-28	1767	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN	...
29	2020-01-29	1480	0.0	NaN	0.0	NaN	NaN	NaN	NaN	NaN	...
...	...	...	...	...	...	...	...	...	...	...	...
305	2020-10-31	542158	157.0	319.0	319.0	98.0	195.0	0.0	3.0	13955.0	...
	2020-										

In [40]:
```
df = df.loc[:,['date','Ecuador']] #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
df
```

Out[40]:

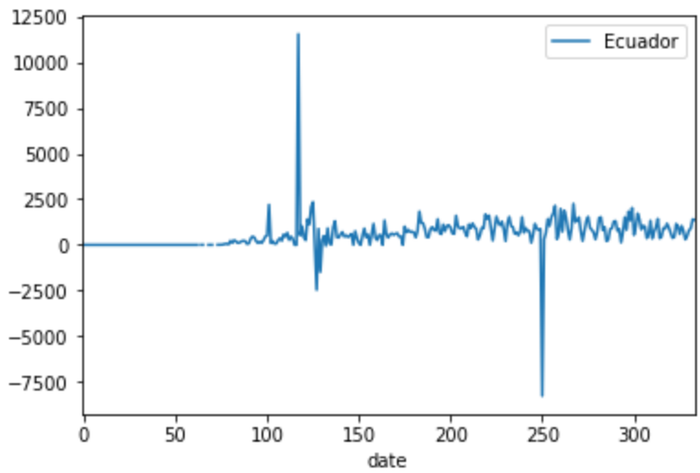
	date	Ecuador
0	-1	0.0
1	0	0.0
2	1	0.0
3	2	0.0
4	3	0.0
5	4	0.0
6	5	0.0
7	6	0.0
8	7	0.0
9	8	0.0
10	9	0.0
11	10	0.0
12	11	0.0
13	12	0.0
14	13	0.0
15	14	0.0
16	15	0.0
17	16	0.0
18	17	0.0
19	18	0.0
20	19	0.0
21	20	0.0
22	21	0.0
23	22	0.0
24	23	0.0
25	24	0.0
26	25	0.0
27	26	0.0
28	27	0.0
29	28	0.0
...	...	...
305	304	845.0
306	305	1045.0
307	306	1002.0
308	307	368.0
309	308	548.0
310	309	1323.0
311	310	350.0
312	311	725.0
313	312	978.0
314	313	1421.0
315	314	362.0
316	315	442.0
317	316	919.0
318	317	883.0

	date	Ecuador
319	318	1161.0
320	319	953.0
321	320	668.0
322	321	381.0
323	322	428.0
324	323	1146.0
325	324	996.0
326	325	594.0
327	326	1036.0
328	327	767.0
329	328	301.0
330	329	492.0
331	330	794.0
332	331	908.0
333	332	1396.0
334	333	1375.0

335 rows × 2 columns

In [41]:
```
df.plot(x ='date', y='Ecuador')
```

Out[41]: <matplotlib.axes._subplots.AxesSubplot at 0x17647402688>



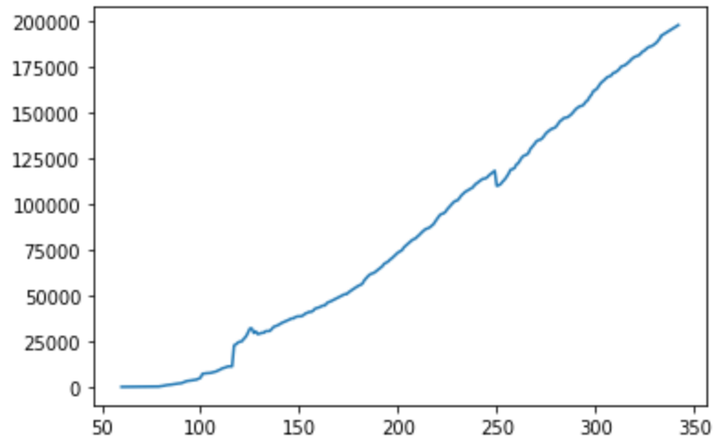
In [42]:
```
filtro = df["Ecuador"][61:]
media = filtro.mean()
mediana = filtro.median()
print('MEDIANA : -->',mediana)
print('MEDIA : -->',media)
```

```
MEDIANA : --> 670.0
MEDIA : --> 709.6988847583643
```

In [43]:
```
url = 'https://covid.ourworldindata.org/data/ecdc/total_cases.csv'
df_t = pd.read_csv(url)
df_t = df_t.replace(np.nan, 0)
FMT = '%Y-%m-%d'
date = df_t['date']
df_t['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("2020-01
df_t = df_t.loc[:,['date','Ecuador']]
y = list(df_t.iloc [:, 1])
x = list(df_t.iloc [:, 0])
prediccion_siguiente = int(y[-1] + mediana)
print(prediccion_siguiente)
```

```
191579
```

In [44]:
```
for i in range(x[-1], x[-1]+10):
    x.append(i)
    y.append(int(y[-1] + mediana))
plt.plot(x[61:], y[61:])
plt.show()
```

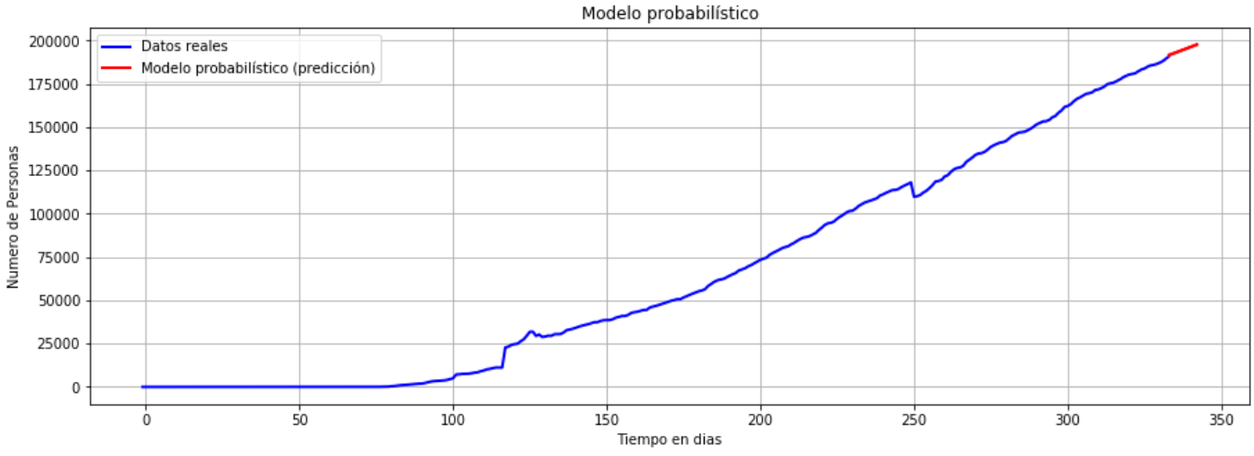


1. Comparar el modelo de predicion matematico vs probabilidad.

Modelo probabilistico

In [45]:
```
fig = plt.figure(figsize=(15,5))
ax = fig.add_subplot(111, axisbelow=True)
ax.plot(x[:len(x)-9], y[:len(x)-9],'b', alpha=1, lw = 2, label = 'Datos reales')
ax.plot(x[len(x)-10:], y[len(x)-10:],'r', alpha=1, lw = 2, label = 'Modelo probabilístico (
ax.set_xlabel('Tiempo en dias')
ax.set_ylabel('Numero de Personas')
ax.set_title("Modelo probabilístico")
ax.legend()
ax.grid()
x_matematico = x[:]
y_matematico = y[:]
ax.plot(x[len(x)-10:], y[len(x)-10:],'r', alpha=1, lw = 2, label = 'Modelo probabilístico (
```

Out[45]:
```
[<matplotlib.lines.Line2D at 0x17646ded8c8>]
```



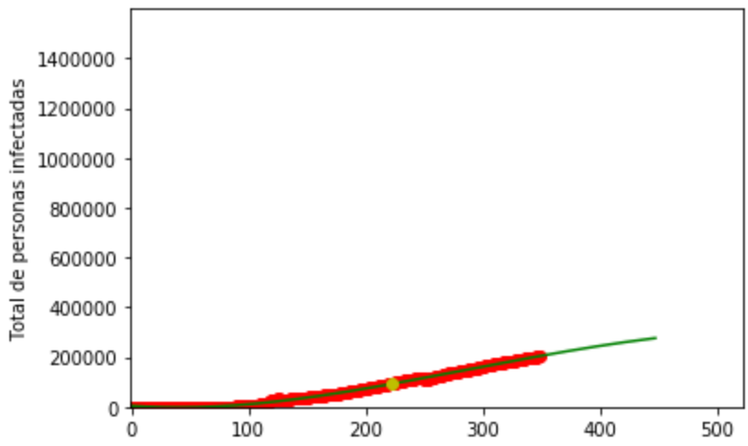
In [46]:
```
prediccion_siguiente = int(y[-1] + mediana)
print('MODELO PROBABILISTICO')
print('TOTAL DE CASOS y MEDIANA: ',prediccion_siguiente)
print('TOTAL, FECHA ACTUAL: ',y[321])
for i in range(x[-1], x[-1]+7):
    x.append(i)
    y.append(int(y[-1] + mediana))
print('ESTIMACION DE CONTAGIOS:',y[-1]+7)
```

```
MODELO PROBABILISTICO
TOTAL DE CASOS y MEDIANA:  198279
TOTAL, FECHA ACTUAL:  180295.0
ESTIMACION DE CONTAGIOS: 202306
```

Modelo polinomial

In [47]:
```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
pf = PolynomialFeatures(degree = 4)
X = pf.fit_transform(np.array(x).reshape(-1, 1))

regresion_lineal = LinearRegression()

regresion_lineal.fit(X, y)
pred_x = list(range(0,max(x)+100))

puntos = pf.fit_transform(np.array(pred_x).reshape(-1, 1))
prediccion_entrenamiento = regresion_lineal.predict(puntos)

print ('CONTAGIOS ACTUALES:', int(prediccion_entrenamiento[222]))

plt.plot(pred_x, prediccion_entrenamiento, color='green')
plt.scatter(x,y,label="Datos Reales",color="red")
plt.ylim((-300,max(y)*7.9))
plt.xlim((min(x)*0.9,max(x)*1.5))
plt.ylabel("Total de personas infectadas")
plt.plot(222,prediccion_entrenamiento[222], 'oy')
plt.show()
```

```
CONTAGIOS ACTUALES: 91916
```

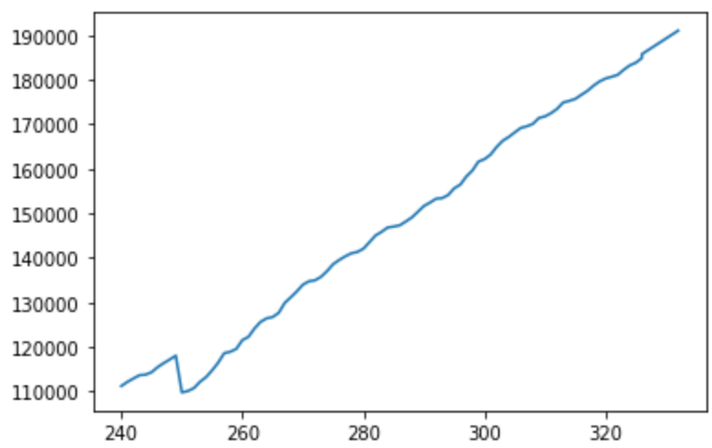


3.Retroceder un semana y comparar el modelo matematico vs probabilidad vs reales. Solo cargan los datos para generar los modelos menos 7 dias.

In [66]:
```
TotalSemanaPasada=df['Ecuador'][241:len(df)-7]
y_semanantes = list(df_t.iloc [:, 1])
x_semanantes = list(df_t.iloc [:, 0])
for i in range (7):
    y_semanantes.pop(-1)
    x_semanantes.pop(-1)
media_semanantes=TotalSemanaPasada.mean()
print ('Una semana antes')
print('Media', media_semanantes)
mediana_semanantes=TotalSemanaPasada.median()
print('Mediana', mediana_semanantes)
prediccion_semana_antes = int(y_semanantes[-1] + mediana_semanantes)
print('Predicion',prediccion_semana_antes)

for i in range(x_semanantes[-1], x_semanantes[-1]+7):
    x_semanantes.append(i)
    y_semanantes.append(int(y_semanantes[-1]+mediana_semanantes))
print('')
print('CASOS POSIBLES')
for i in range(7):
    print('DIA',i+1,':',round(prediccion_semana_antes))
    prediccion_semana_antes=prediccion_semana_antes+mediana_semanantes

plt.plot(x_semanantes[241:], y_semanantes[241:])
plt.show
```

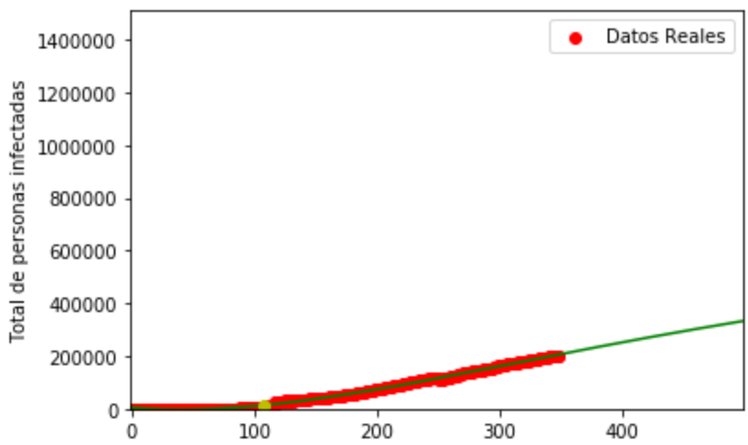
```
Una semana antes
Media 854.3333333333334
Mediana 883.0
Predicion 185759

CASOS POSIBLES
DIA 1 : 185759
DIA 2 : 186642
DIA 3 : 187525
DIA 4 : 188408
DIA 5 : 189291
DIA 6 : 190174
DIA 7 : 191057
```

Out[66]:
```
<function matplotlib.pyplot.show(*args, **kw)>
```



In [68]:
```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures

pf = PolynomialFeatures(degree = 4)
X = pf.fit_transform(np.array(x_semanantes).reshape(-1, 1))

regresion_lineal = LinearRegression()
regresion_lineal.fit(X,y_semanantes )

pred_x = list(range(0,max(x_semanantes)+200))
puntos = pf.fit_transform(np.array(pred_x).reshape(-1, 1))
prediccion_entrenamiento = regresion_lineal.predict(puntos)
print ('Predicción una semana antes:', round(prediccion_entrenamiento[108]), 'contagiados')
plt.plot(pred_x, prediccion_entrenamiento, color='green')
plt.scatter(x,y,label="Datos Reales",color="red")
plt.ylim((-300,max(y_semanantes)*7.9))
plt.xlim((min(x_semanantes)*0.9,max(x_semanantes)*1.5))
plt.ylabel("Total de personas infectadas")
plt.plot(108,prediccion_entrenamiento[108], 'oy')
plt.legend()
plt.show()
```

```
Predicción una semana antes: 13691.0 contagiados
```



Conclusion

Una modelo probabilistico nos ayudara a preveer el numero de contagios posibles en los siguientes dias de una manera mas acertda