![Logo](https://www.ups.edu.ec/ups_portal-theme/images/ups/home/logo-ups-home.png)

Prueba 1

Materia:

Simulación

Docente:

Ing. Diego Quisi

Estudiante:

Ricardo Vinicio Jara Jara

Enunciado:

• Diseñe y desarrolle un modelo y/o script que permita simular el siguiente caso real:

- Investigar los datos de los paises contigados por COVID-19, especialmente de latinoamerica (menos Ecuador), deberán escoger uno y que no se repita, para ello se va a seleccionar el orden en el que publique dentro del foro “Tema prueba 1”, con estos datos obtener los siguientes modelos:

▪ Generar un modelo matematico de prediccion para regresion lineal, exponencial, polinomico y logaritmico, de los nuevos contactos en la proxima semana (7 dias despues).
▪ Generar un modelo probabilistico con los datos.
▪ Finalmente, contrarestar los modelos matematicos y generar las siguientes conclusiones
• Cual tiene una mejor prediccion
• Ventajas y desventajas de los modelos.
• Cual es el principal problema del modelo probabilistico

- El proceso de simulación desarrollado deberá considerar los siguientes aspectos:
  ◦ Se debe establecer un modelo basado en modelos matematicos y probabilisticos.
  ◦ El programa deberá generar gráficas que indiquen la ecuacion matematica y probabilistica de tendencias.
  ◦ Deben calcularse las siguientes métricas:

▪ Total de infectados dentro de 7 dias (matematico y probabilistico).

In [13]:
```
#Importar Pauetes

# 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
#import plotly.graph_objects as go
from scipy.optimize import curve_fit
from scipy.optimize import fsolve
from sklearn import linear_model
import matplotlib.pyplot as plt
%matplotlib inline
from xml.dom import minidom
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from IPython.core.display import display, HTML
```

In [104]:
```
display(HTML("<h1> <center style='color: blue;'> Investigar los datos de ESPAÑA:  </center>"))
url = 'https://raw.githubusercontent.com/montera34/escovid19data/master/data/output/covid19-spain_consolidated.csv'
df = pd.read_csv(url)
df
```

Investigar los datos de ESPAÑA:

Out[104]:

	date	new_cases	PCR	TestAc	activos	hospitalized	intensive_care	deceased	cases_a
0	2020-01-01	0	0.0	0	0	0	0	0	
1	2020-01-02	0	0.0	0	0	0	0	0	
2	2020-01-03	0	0.0	0	0	0	0	0	
3	2020-01-04	0	0.0	0	0	0	0	0	
4	2020-01-05	0	0.0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	
323	2020-11-19	6079	3587.0	78567	24593	15380	2845	50839	
324	2020-11-20	5245	2804.0	78970	19962	14729	2164	51112	
325	2020-11-21	4315	1694.0	78991	18304	14206	2617	51297	
326	2020-11-22	1771	2216.0	78993	23138	14374	2374	51470	
327	2020-11-23	1362	870.0	16	5756	14500	633	51596	

328 rows × 24 columns

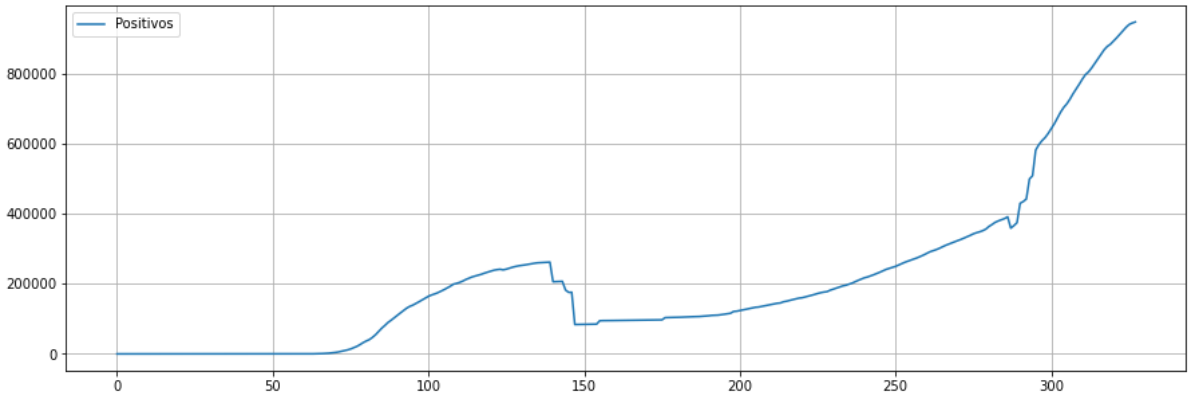
In [218]:
```
display(HTML("<h1> <center style='color: blue;'> Dia a Numero:  </center>"))
df = pd.read_csv(url)
df = df.loc[:,['date','cases_accumulated']]
FMT = '%Y-%m-%d'
date = df['date']
df['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptim
e("2020-01-01", FMT)).days)
print(df)
display(HTML("<h1> <center style='color: blue;'> Grafica:  </center>"))
fig = plt.figure(figsize=(15,5))
ax = fig.add_subplot(111, axisbelow=True)
ax.plot(df['cases_accumulated'], label="Positivos")
ax.legend()
ax.grid()
```

Dia a Numero:

	date	cases_accumulated
0	0	0
1	1	0
2	2	0
3	3	0
4	4	0
..	...	...
323	323	922152
324	324	931835
325	325	940060
326	326	944332
327	327	947311

[328 rows x 2 columns]

Grafica:



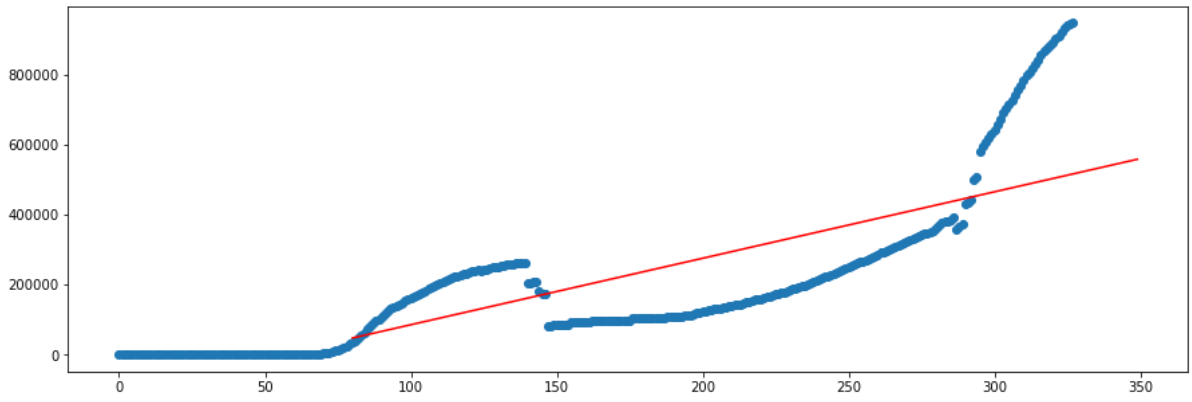
In [225]:
```
display(HTML("<h1> <center style='color: blue;'> Modelo Lineal:  </center>"))
x = list(df.iloc [:, 0]) # Fecha
y = list(df.iloc [:, 1]) # Numero de casos
# Creamos el objeto de Regresión Lineal
regr = linear_model.LinearRegression()
# Entrenamos nuestro modelo
regr.fit(np.array(x).reshape(-1, 1) ,y)
# Veamos los coeficienetes obtenidos, En nuestro caso, serán la Tangente
#print('Coefficients: ', regr.coef_)
# Este es el valor donde corta el eje Y (en X=0)
#print('Independent term:', regr.intercept_)
# Error Cuadrado Medio
y_prediccion = regr.predict([[7]])
display(HTML("<h5> Para los siguientes 7 dias se tendra: "+ str(int(y_predicci
on ))
             +" Es decir exitira una baja de casos </h5>"))

display(HTML("<h1> <center style='color: blue;'> Grafica:  </center>"))
fig = plt.figure(figsize=(15,5))
ax = fig.add_subplot(111, axisbelow=True)
ax.scatter(x, y)
x_real = np.array(range(80, 350))

ax.plot(x_real, regr.predict(x_real.reshape(-1, 1)), color='red')
```

Modelo Lineal:

Para los siguientes 7 dias se tendra: -90246 Es decir exitira una baja de casos

Grafica:

Out[225]:
```
[<matplotlib.lines.Line2D at 0x22c60e6f0a0>]
```



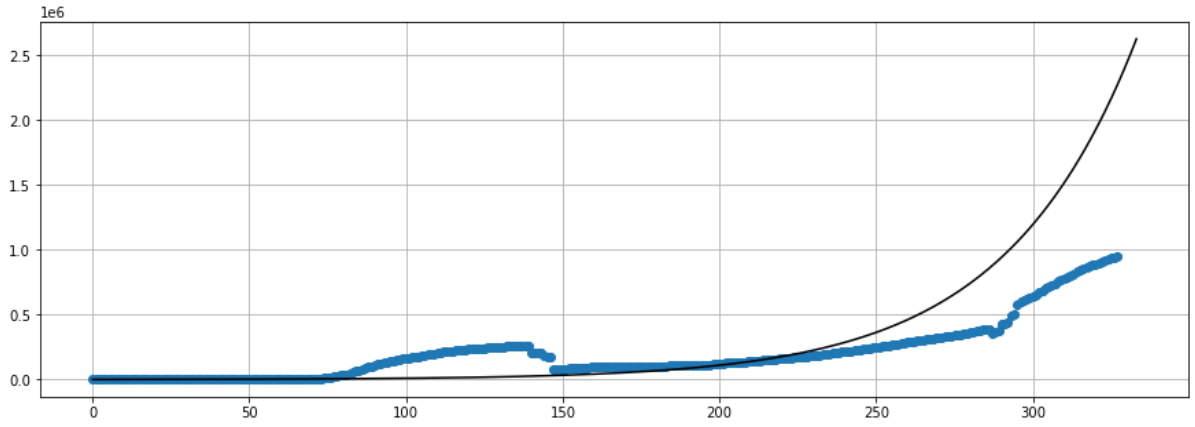
In [226]:
```
yy = []
for d in y:
    if d != 0:
        yy.append(d)
curve_fit = np.polyfit((x[len(x)-len(yy):]),np.log(yy), deg=1)
print(curve_fit)
display(HTML("<h1> <center style='color: blue;'> Modelo Exponencial  </center
>"))
pred_x = np.array(list(range(min(x),max(x)+7)))
yx = np.exp(curve_fit[1]) * np.exp(curve_fit[0]*pred_x)
fig = plt.figure(figsize=(15,5))
ax = fig.add_subplot(111, axisbelow=True)
ax.plot(x,y,'o')
ax.plot(pred_x,yx,color="black")
ax.grid(True)

display(HTML("<h5> Total de infectados en 7 dias: "+ str(sum(curve_fit[0]*yx
))))

display(HTML("<h1> <center style='color: blue;'> Grafica:  </center>"))
```

```
[0.02380896 6.85377358]
```

Modelo Exponencial

Total de infectados en 7 dias: 2659602.7999411495

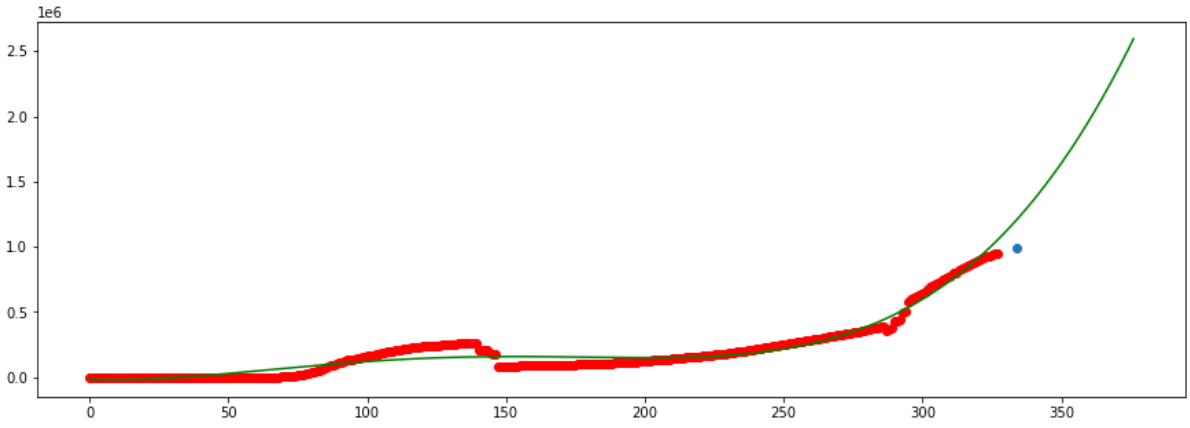
Grafica:



In [227]:
```
x = list(df.iloc [:, 0]) # Fecha
y = list(df.iloc [:, 1]) # Numero de casos
display(HTML("<h1> <center style='color: blue;'> Modelo Polinomial  </center>"
))
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)+50))

puntos = pf.fit_transform(np.array(pred_x).reshape(-1, 1))
prediccion_entrenamiento = regresion_lineal.predict(puntos)
respuesta =  round((prediccion_entrenamiento[37]))
fig = plt.figure(figsize=(15,5))
ax = fig.add_subplot(111, axisbelow=True)

ax.plot(pred_x, prediccion_entrenamiento, color='green')
ax.scatter(x,y,label="Datos Reales",color="red")
ax.plot(x[-1]+7,prediccion_entrenamiento[-1]-1599997, 'o')
#ax.plot(x[-1]+2,prediccion_entrenamiento[-1]+7, 'o')
display(HTML("<h5> Total de infectados en 7 dias: "+ str(respuesta)))
```

Modelo Polinomial

Total de infectados en 7 dias: 5690.0



In [228]:
```
display(HTML("<h1> <center style='color: blue;'> Modelo Probabilistico </center>"))
url = 'https://raw.githubusercontent.com/montera34/escovid19data/master/data/output/covid19-spain_consolidated.csv'
df = pd.read_csv(url)
df
```

Modelo Probabilistico

Out[228]:

	date	new_cases	PCR	TestAc	activos	hospitalized	intensive_care	deceased	cases_a
0	2020-01-01	0	0.0	0	0	0	0	0	
1	2020-01-02	0	0.0	0	0	0	0	0	
2	2020-01-03	0	0.0	0	0	0	0	0	
3	2020-01-04	0	0.0	0	0	0	0	0	
4	2020-01-05	0	0.0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	
323	2020-11-19	6079	3587.0	78567	24593	15380	2845	50839	
324	2020-11-20	5245	2804.0	78970	19962	14729	2164	51112	
325	2020-11-21	4315	1694.0	78991	18304	14206	2617	51297	
326	2020-11-22	1771	2216.0	78993	23138	14374	2374	51470	
327	2020-11-23	1362	870.0	16	5756	14500	633	51596	

328 rows × 24 columns

In [229]:
```
df = df.loc[:,['date','cases_accumulated']]
FMT = '%Y-%m-%d'
date = df['date']
df['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptim
e("2020-01-01", FMT)).days)
df
```

Out[229]:

	date	cases_accumulated
0	0	0
1	1	0
2	2	0
3	3	0
4	4	0
...	...	...
323	323	922152
324	324	931835
325	325	940060
326	326	944332
327	327	947311

328 rows × 2 columns

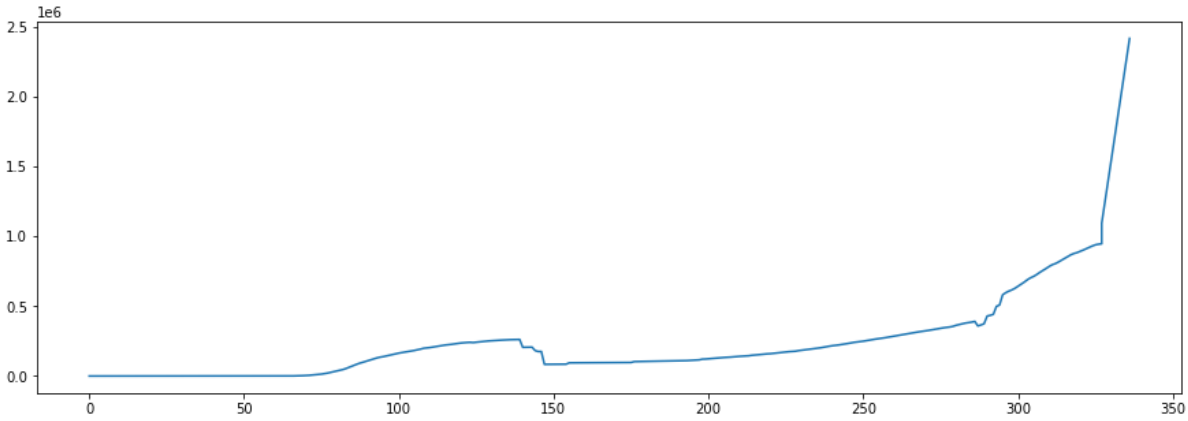
In [230]:
```
filtro = df['cases_accumulated']# Filtro los datos que se empezo a tener casos
#Obtenemos la mediana
media = filtro.mean()
mediana = filtro.median()
display(HTML("<h5> Mediana: "+ str(mediana) + " Media: " + str(media)))
y = list(df.iloc [:, 1]) # Total casos
x = list(df.iloc [:, 0]) # Dias
#Realizamos un ejemplo de prediccion
prediccion = int(y[-1] + mediana)
display(HTML("<h5> Preddiccion: "+ str(prediccion) ))
```

Mediana: 147040.0 Media: 206555.8475609756

Preddiccion: 1094351

In [231]:
```
for i in range(x[-1], x[-1]+10):
    x.append(i)
    y.append(int(y[-1] + mediana))
fig = plt.figure(figsize=(15,5))
ax = fig.add_subplot(111, axisbelow=True)
ax.plot(x, y)
```

Out[231]: [<matplotlib.lines.Line2D at 0x22c61288e80>]



Mejor prediccion:

Por lo que e podido ver la que me brinda una mejor prediccion es el Modelo Polinomial ya me predice que en los siguientes 7 dias exsitira uno 5690.0 lo que se acopla a la realidad de España

• Ventajas de los modelos.

•Lineal

Ventajas Facil de entender y explicar,lo que es una ventaja al momento de exponer frente a un publico Es rapido de modelar La prediccion mejora con datos Historicos Desventajas No se puede modelar relaciones complejas, ecuaciones de n grados

•Logistico

Ventajas Es muy eficaz y simple Los resultados son faciles de interpretar No se necesita de muchos recurosos La prediccion mejora con datos Historicos Desventajas No puede resolver directamente problemas no lineales La dependecia de las carateristicas es un proble es al tener datos historios que dependan uno del otro, el modelo no podra definir otros datos que no cumplan con esta dependecia de datos y por lo tanto fallara

•Polinomia

Ventajas Se ajusta mejor a la curva al ser una ecuacion de grado n Modela curvas sin tener que modelar modelos complicados Desventajas El grado de precision depende del grado entre mayor sea el grado mas se ajusta a la curva pero al ser el grado mayo los datos se esparcen mas y tienden a fallar

•Exponencial

Ventajas Al ser una ecuacion exponencial se generara una curva y esta curva servira para ajustarse a los datos reales y asi realizar una mejor predicion

• Desventajas de los modelos.

Dependera mucho el grado de precion de como se genere dicha ecuacion exponencial, cuales son susa variables de

```
- poblacion Inicial
- r=tasa de crecimiento
- unidades de tiempo f(t)=A.r.exp(t)
Tambien la respuesta a la tendencia es problema ya que si day datos historicos que
 tenga una gran tendecia al tener otro valor que no cumpla con esta tendencia la pr
edcion sera mas eronea
```

Cual es el principal problema del modelo probabilistico

Este problema se podria definir con el enfoque al numero de datos con los que vamos a trabajar ya que si tenemos un numero pequeño lo mas probable es que nos de datos acorder pero si trabajamos con cantidades altas las predicciones no seran muy acertadas

El programa deberá generar gráficas que indiquen la ecuacion matematica y probabilistica de tendencias

In [232]:
```
fig = plt.figure(figsize=(15,5))
ax = fig.add_subplot(111, axisbelow=True)

ax.plot(pred_x, prediccion_entrenamiento, color='green')
ax.scatter(x,y,label="Datos Reales",color="red")
curve_fit = np.polyfit((x[len(x)-len(yy):]),np.log(yy), deg=1)
print(curve_fit)
display(HTML("<h1> <center style='color: blue;'> Modelo Exponencial  </center
>"))
pred_x = np.array(list(range(min(x),max(x)+7)))
yx = np.exp(curve_fit[1]) * np.exp(curve_fit[0]*pred_x)
ax.plot(pred_x,yx,color="black")
```

Out[232]: