UNIVERSIDAD POLITECNICA SALESIANA

SEDE CUENCA

CARRERA: *INGENIERIA DE SISTEMAS*

Nombre: Bryam Gabriel Mora Lituma

Materia: Simulacion

Fecha:02/05/2021

In [1]:

```python
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 [8]:

```python
url = "datos.csv"
df = pd.read_csv(url,sep="\;")
```

```
C:\Users\Bryam\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: ParserWarning: Falling back to the 'pyt
hon' engine because the 'c' engine does not support regex separators (separators > 1 char and different fro
m '\s+' are interpreted as regex); you can avoid this warning by specifying engine='python'.
```

In [9]:

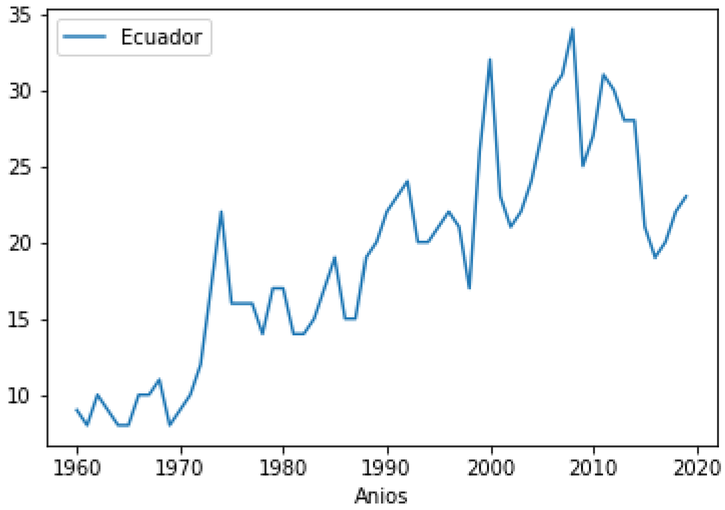
```
df = df[df['Name'].isin(['Ecuador'])]

df = df.loc[:,['Name','1960','1961','1962','1963','1964','1965','1966','1967','1968',\
           '1968','1969','1970','1971','1972','1973','1974','1975','1976',\
           '1977','1978','1979','1980','1981','1982','1983','1984','1985',\
           '1986','1987','1988','1989','1990','1991','1992','1993','1994',\
           '1995','1996','1997','1998','1999','2000','2001','2002','2003',\
           '2004','2005','2006','2007','2008','2009','2010','2011','2012',\
           '2013','2014','2015','2016','2017','2018','2019']]
df = df.set_index('Name').T

ecua = df["Ecuador"].astype(int)
df['Ecuador'] = ecua
anios = df['Ecuador'].index.tolist()
df['Anios'] = anios
an = df["Anios"].astype(int)
df["Anios"] = an
```

In [10]:

```
df.plot(y="Ecuador", x="Anios")
```

Out[10]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f0c73bb308>
```



In [11]:

```python
x = list(df.iloc [:, 0])
y = list(df.iloc [:, 1])

regr = linear_model.LinearRegression()

regr.fit(np.array(x).reshape(-1, 1), y)

print('Coeficintes: \n',regr.coef_)

print("Independiente termino: \n",regr.intercept_)
```

```
Coeficintes: 
 [2.13272535]
Independiente termino: 
 1948.9404237135661
```

In [12]:

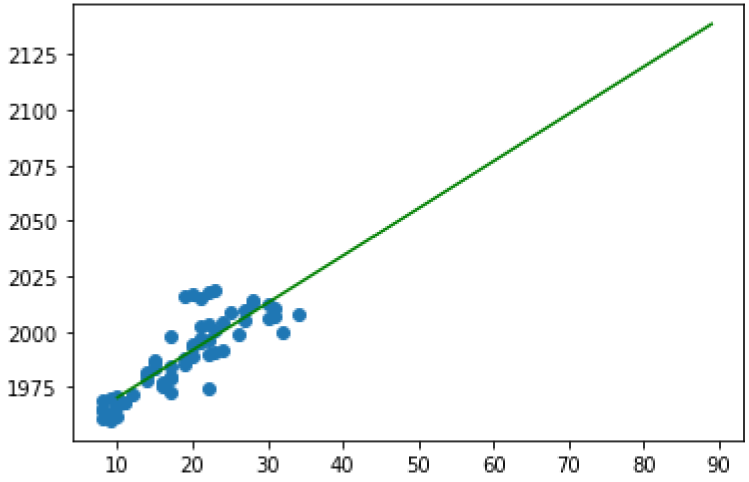
```python
y_prediccion = regr.predict([[100]])
print(int(y_prediccion))
```

```
2162
```

In [13]:

```
plt.scatter(x,y)
x_real = np.array(range(10,90))
print(x_real)
plt.plot(x_real, regr.predict(x_real.reshape(-1,1)), color='green')
plt.show()
```

```
[10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 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]
```



Modelo Logistico

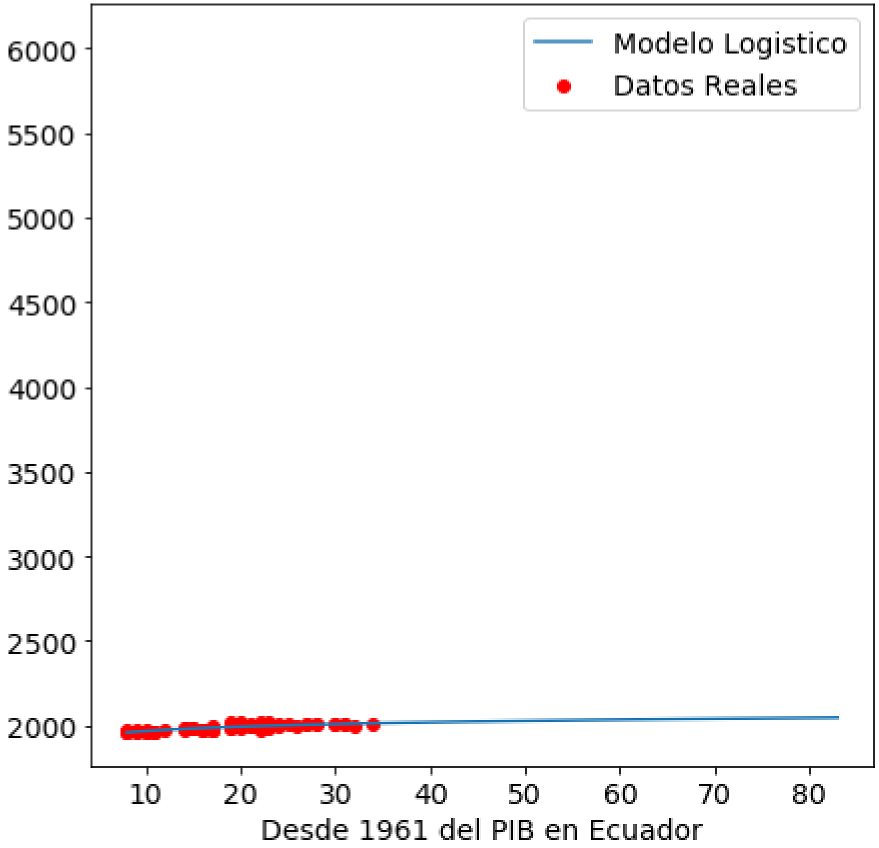
In [14]:

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

exp_fit = curve_fit(modelo_logistico,x,y)
print(exp_fit)
```

```
(array([1882.39851097,   37.3091618 ]), array([[ 66.61474569, -22.82928999],
       [-22.82928999,   7.97891718]]))
```

In [15]:

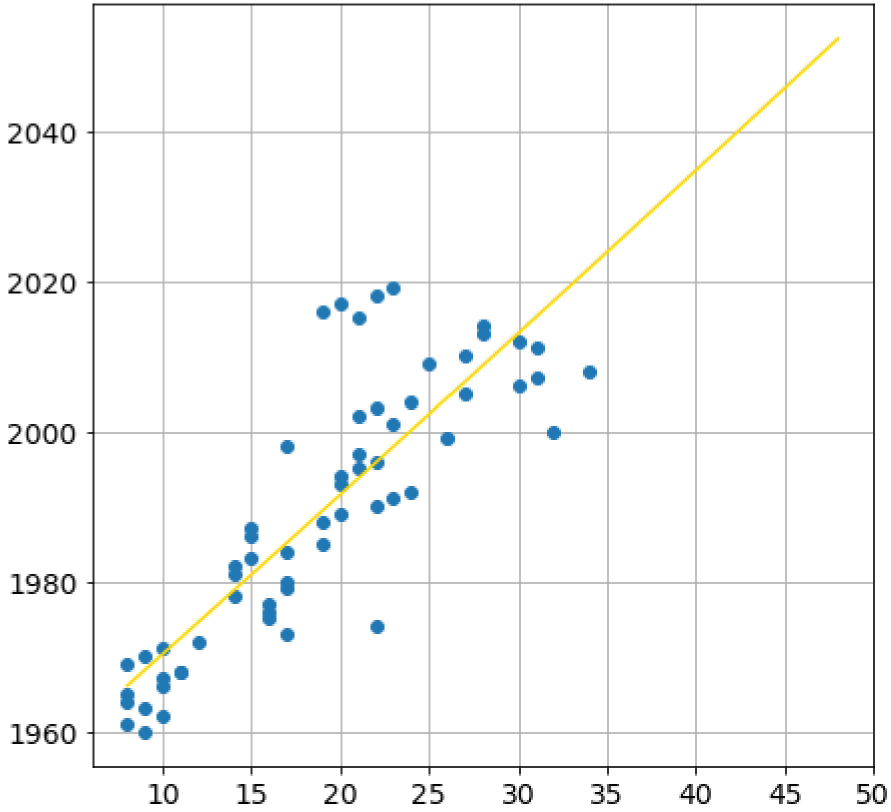
```python
pred_x = list(range(min(x),max(x)+50)) # Predecir 50 dias mas
plt.rcParams['figure.figsize'] = [7, 7]
plt.rc('font', size=14)
# Real data
plt.scatter(x,y,label="Datos Reales",color="red")
# Predicted exponential curve
plt.plot(pred_x, [modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i in pred_x], label="Modelo Logist
plt.legend()
plt.xlabel("Desde 1961 del PIB en Ecuador")
plt.ylabel("")
plt.ylim((min(y)*0.9,max(y)*3.1)) # Definir los limites de Y
plt.show()
```



In [16]:

```python
curve_fit = np.polyfit(x, np.log(y), deg=1)
print(curve_fit)
pred_x = np.array(list(range(min(x), max(x)+15)))
yx = np.exp(curve_fit[1]) * np.exp(curve_fit[0]*pred_x)
plt.plot(x,y,"o")
plt.plot(pred_x,yx, color="gold")
plt.grid(True)
```

```
[1.07303055e-03 7.57519397e+00]
```



In [ ]:

```python

```