

Nombre: Jorge Arévalo

Docente: Ing. Diego Quisi

Materia: Simulación

Prueba Práctica

In [1]:

```
1 # Importar las librerías para el analisis
2 import pandas as pd
3 import numpy as np
4 from datetime import datetime, timedelta
5 from sklearn.metrics import mean_squared_error
6 from scipy.optimize import curve_fit
7 from scipy.optimize import fsolve
8 from sklearn import linear_model
9 import matplotlib.pyplot as plt
10 %matplotlib inline
```

In [2]:

```
1 url = 'owid-covid-data.csv'
2 df = pd.read_csv(url)
3 df
```

Out[2]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed
0	ABW	North America	Aruba	2020-03-13	2.0	2.0	NaN
1	ABW	North America	Aruba	2020-03-19	NaN	NaN	0.28
2	ABW	North America	Aruba	2020-03-20	4.0	2.0	0.28
3	ABW	North America	Aruba	2020-03-21	NaN	NaN	0.28
4	ABW	North America	Aruba	2020-03-22	NaN	NaN	0.28
...
58693	NaN	NaN	International	2020-11-19	696.0	NaN	NaN
58694	NaN	NaN	International	2020-11-20	696.0	NaN	NaN
58695	NaN	NaN	International	2020-11-21	696.0	NaN	NaN
58696	NaN	NaN	International	2020-11-22	696.0	NaN	NaN
58697	NaN	NaN	International	2020-11-23	696.0	NaN	NaN

58698 rows × 50 columns

In [3]:

```

1 df = df[df['location'].isin(['Costa Rica'])] #Filtro la Informacion solo para Ecuador
2 df = df.loc[:,['date','total_cases']] #Selecciono las columnas de analisis
3 # Expresar las fechas en numero de dias desde el 01 Enero
4 FMT = '%Y-%m-%d'
5 #FMT = '%m/%d/%Y'
6 date = df['date']
7 df['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptime("2020-0
8 df

```

Out[3]:

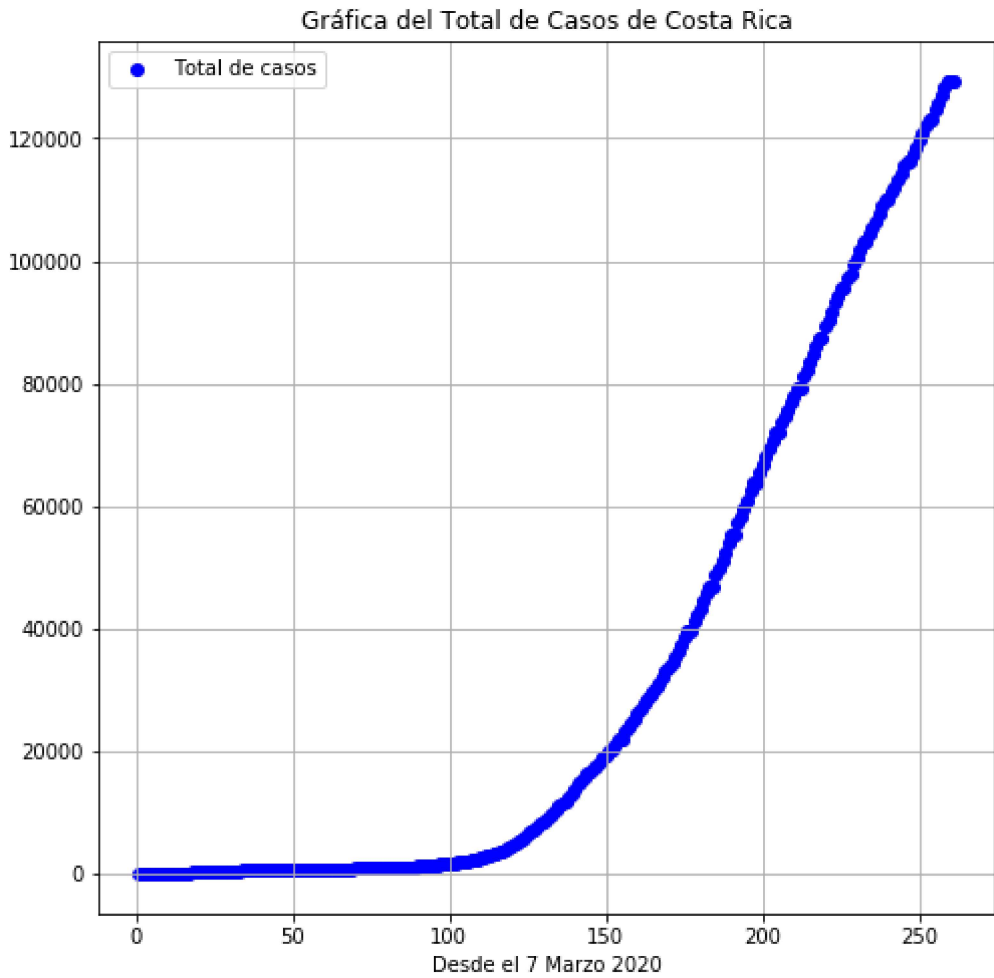
	date	total_cases
12074	0	1.0
12075	2	5.0
12076	3	9.0
12077	4	13.0
12078	5	22.0
...
12330	257	127012.0
12331	258	128231.0
12332	259	129418.0
12333	260	129418.0
12334	261	129418.0

261 rows × 2 columns

Regresión Lineal

In [30]:

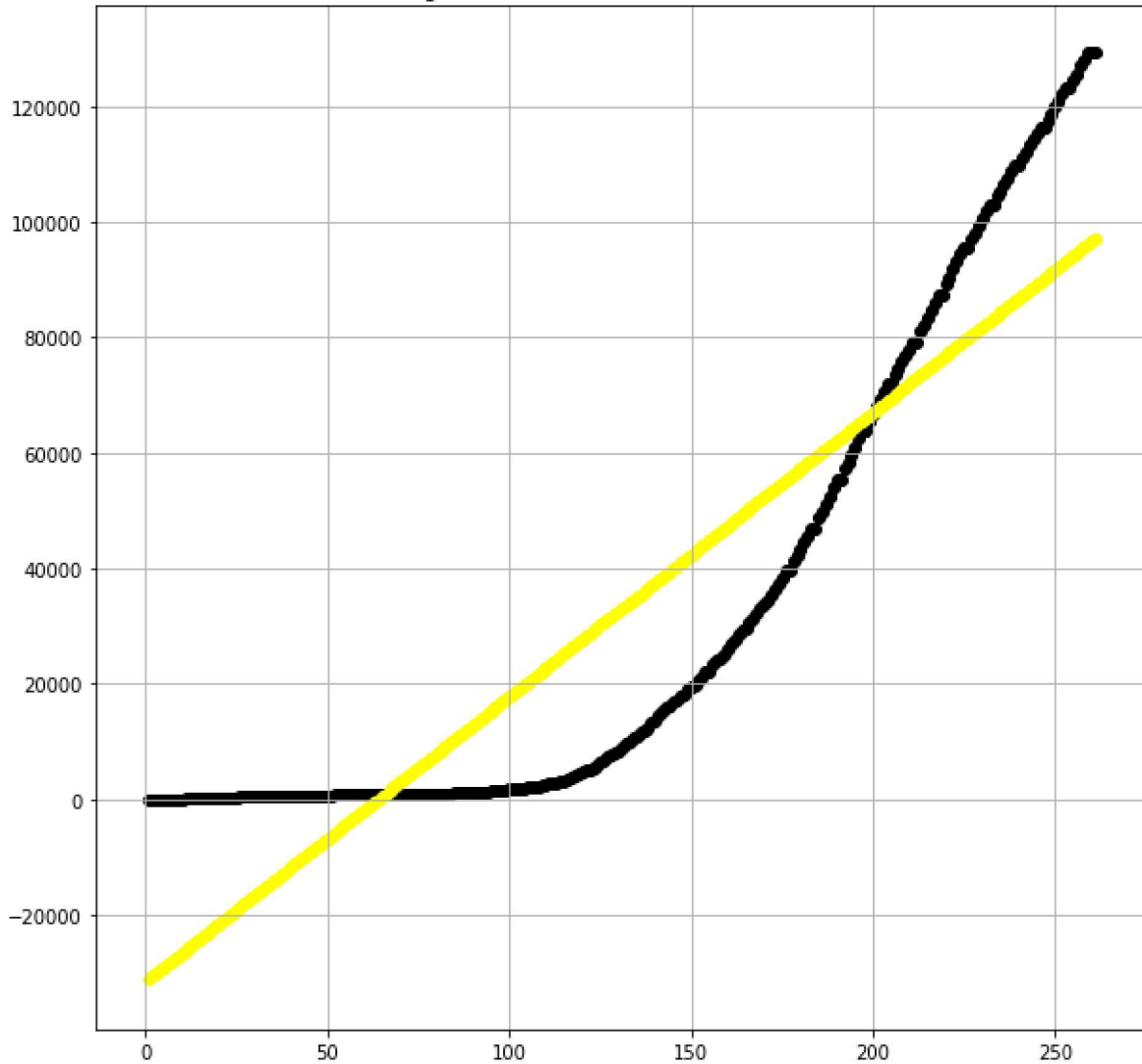
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from sklearn.linear_model import LinearRegression
4 import pandas as pd
5 from datetime import datetime, timedelta
6 import matplotlib.pyplot as plt
7 %matplotlib inline
8
9 def f(x):
10     np.random.seed(42)
11     y = 0.1*x + 1.25 + 0.2*np.random.randn(x.shape[0])
12     return y
13
14 df = pd.read_csv('owid-covid-data.csv').fillna(0)
15 ndf= df.loc[(df['location'] == 'Costa Rica') & (df['total_cases'] != 0)]
16 ndf1=ndf[['date', 'total_cases']]
17
18 x=np.arange(1,len(ndf1)+1,1)
19 y=np.array(ndf1.values[:,1])
20 #z=np.array(ndf1.values[:,2])
21
22
23 plt.figure(figsize=(8, 8))
24 plt.scatter(x,y,label='Total de casos', color='Blue')
25 #plt.scatter(x,z,label='Número de muertes', color='Red')
26 plt.grid(True)
27 plt.legend()
28 plt.title('Gráfica del Total de Casos de Costa Rica');
29 plt.xlabel("Desde el 7 Marzo 2020");
```



In [31]:

```
1  # Creamos el objeto de Regresión Lineal
2  regr = LinearRegression()
3  # Entrenamos nuestro modelo
4  #regr.fit(np.array(x).reshape(-1, 1), y)
5  regr.fit(x.reshape((-1, 1)), y)
6  #pred = regr.predict(np.array(x).reshape(-1, 1))
7  pred = regr.predict(x.reshape((-1, 1)))
8
9  plt.figure(figsize=(10, 10))
10 plt.scatter(x, y, color='black')
11 plt.grid(True)
12 plt.title('Regresión Lineal del Total de infectados');
13 plt.scatter(x, pred, color='yellow')
14 #plt.legend()
15 plt.show()
16
17 if (regr.intercept_ < 0):
18     ecua='y = {}x {}'
19 else:
20     ecua='y = {}x + {}'
21 print('Ecuación: ', ecua.format(regr.coef_[0], regr.intercept_))
22
23 #Número 7 del día de la cuantos contagiados hay por el covid
24 num = 268
25 new = np.array([int(num)])
26 prediccion = regr.predict(new.reshape(-1, 1))
27 print("La predicción de", num, "es: " , prediccion[0])
28
```

Regresión Lineal del Total de infectados



Ecuación: $y = 492.733540540358x - 31498.88308281754$

La predicción de 268 es: 100553.7057819984

----- **Regresión Exponencial** -----

In [32]:

```

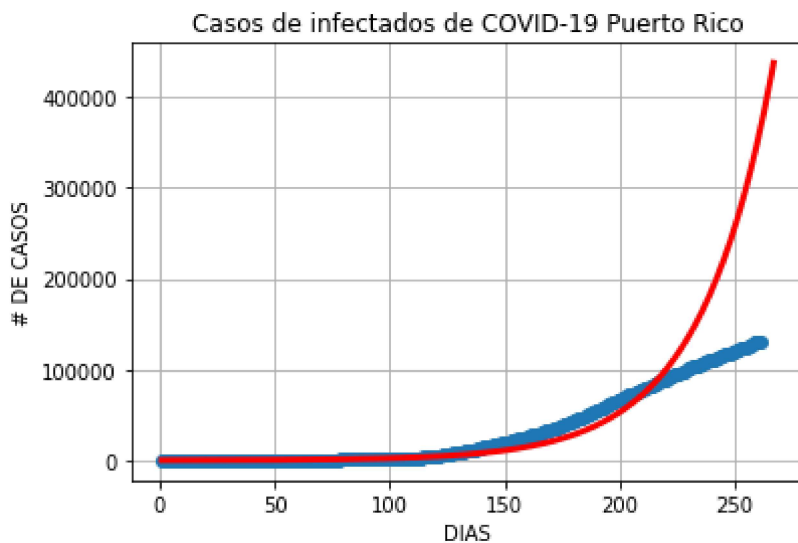
1 df = pd.read_csv('owid-covid-data.csv').fillna(0) # poniendo datos nan a cero
2 ndf= df.loc[(df['location'] == 'Costa Rica') & (df['total_cases'] != 0)] # filtrando po
3 ndf1=ndf[['date', 'total_cases']]
4 x=np.arange(1,len(ndf1)+1,1) # arreglo de x lo creo para simular el numero del dia y el
5 y=np.array(ndf1.values[:,1], dtype='float')
6 xexp=x
7 yexp=y
8 ndf1exp=ndf1
9
10 curve_fit=np.polyfit(xexp,np.log(yexp),1)
11 print(curve_fit)
12 pred_xe=np.array(list(range(min(xexp),max(xexp)+7)))
13 yx=np.exp(curve_fit[1])*np.exp(curve_fit[0]*pred_xe)
14 plt.title('Casos de infectados de COVID-19 Puerto Rico')
15 plt.plot(xexp,yexp,"o")
16 plt.plot(pred_xe,yx,color='r',linewidth=3.0)
17 plt.xlabel('DIAS')
18 plt.ylabel('# DE CASOS')
19 plt.grid(True)
20 print(yx[len(yx)-1])

```

```

[0.0315648  4.56078712]
437393.7570901771

```



----- Regresión Polinomial -----

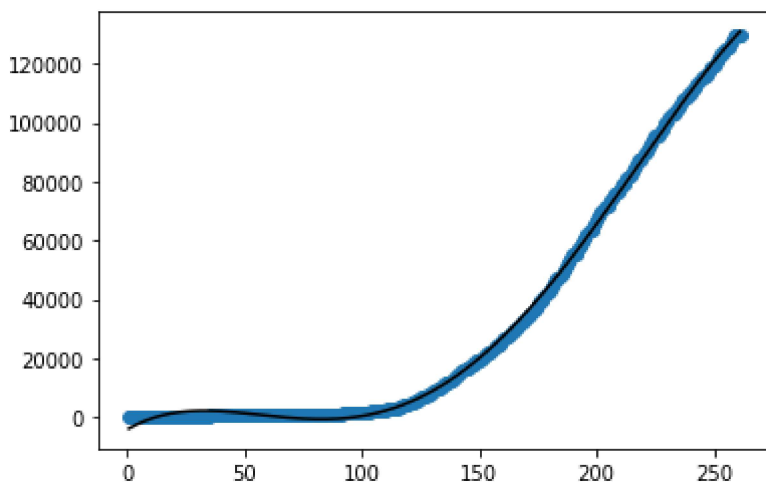
In [33]:

```

1 df = pd.read_csv('owid-covid-data.csv').fillna(0)
2 ndf= df.loc[(df['location'] == 'Costa Rica') & (df['total_cases'] != 0)]
3 ndf1=ndf[['date', 'total_cases']]
4 x=np.arange(1,len(ndf1)+1,1, dtype='float')
5 y=np.array(ndf1.values[:,1], dtype='float')
6 fun1 = np.poly1d(np.polyfit(x, y, 4))
7 print(fun1)
8 plt.scatter(x, y)
9 plt.plot(x, fun1(x), c='black')
10 plt.show()
11
12 print("Predicción: ",fun1(len(x)+7),2)

```

4 3 2
 $-0.0001454 x^4 + 0.07995 x^3 - 10.79 x^2 + 471.7 x - 4260$



Predicción: 136360.35162643762 2

----- **Regresión Logarítmica** -----

In [19]:

```
1 from scipy.optimize import curve_fit
2 from sklearn.linear_model import LogisticRegression
3 def modelo_logistico(x,a,b):
4     return a+b*np.log(x)
5
6 exp_fit = curve_fit(modelo_logistico,x,y)
7 #exp_fit1 = curve_fit(modelo_logistico,x,z)
8
9 #print(exp_fit)
10 print(" ")
11 #print(exp_fit1)
```

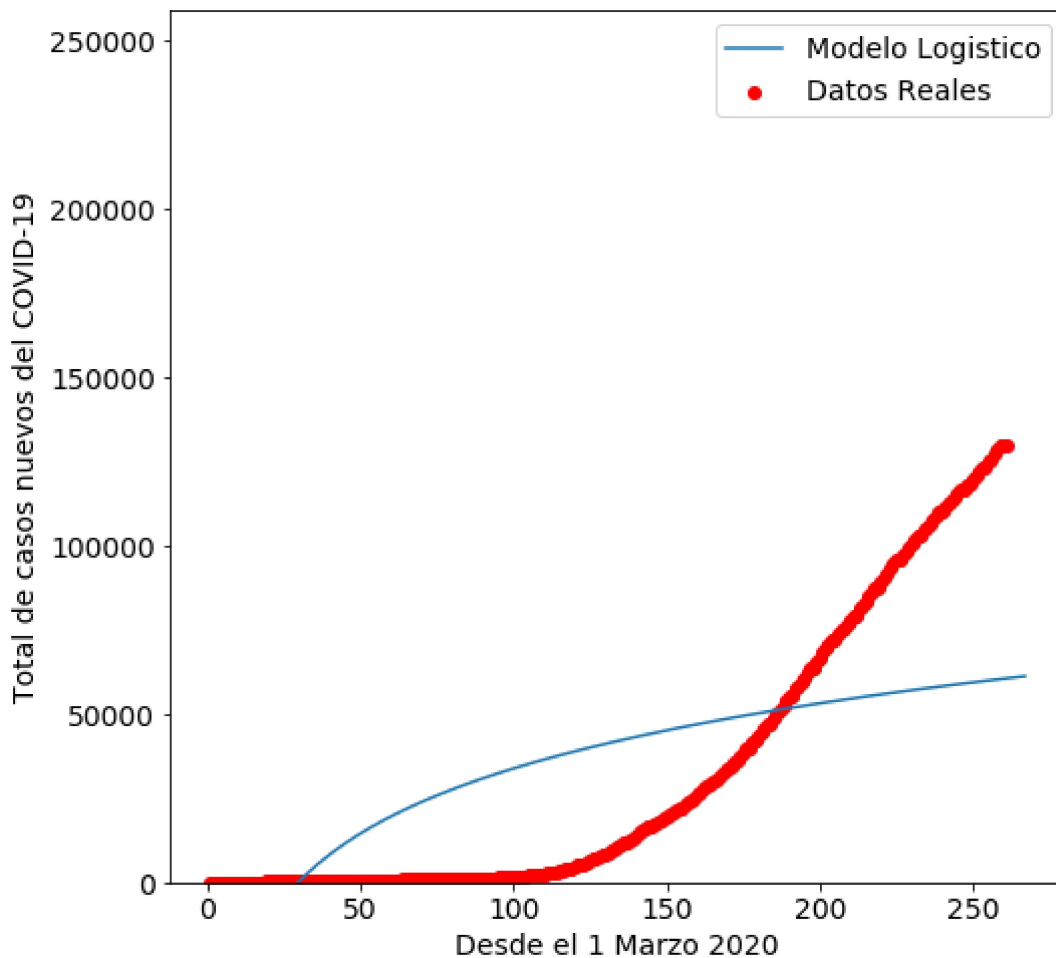
```
(array([-94531.1262067 ,  27863.86270053]), array([[ 91583411.77474916, -191
59453.00896827],
          [-19159453.00896827,  4184472.13733146]]))
```

In [28]:

```

1 pred_x = list(range(min(x),max(x)+7)) # Predecir 50 dias mas
2 plt.rcParams['figure.figsize'] = [8, 8]
3 plt.rc('font', size=14)
4 # Real data
5 plt.scatter(x,y,label="Datos Reales",color="red")
6 val=[modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i in pred_x]
7 plt.plot(pred_x, [modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i in pred_x], label="Modelo Logistico")
8 plt.legend()
9 plt.xlabel("Desde el 1 Marzo 2020")
10 plt.ylabel("Total de casos nuevos del COVID-19")
11 plt.ylim(0,max(y)*2)
12 plt.show()
13
14 print("Predicción: ",val[len(pred_x)-1])

```



Predicción: 61151.203284706295

Finalmente, contrarrestar los modelos matematicos y generar las siguientes conclusiones

Cual tiene una mejor prediccion

El que tiene mejor la predicción es el modelo polinomial.

Ventajas y desventajas de los modelos.

Modelo Lineal

Ventajas

- Sencillo y claro.
- Trabaja con cualquier tamaño de muestra.

Desventajas

- Asume que los datos se comportan de forma lineal.
- Modela las relaciones entre variables dependientes e independientes que son lineales.

Modelo Logaritmico

Ventajas

- No es necesario disponer de grandes recursos computacionales, tanto en entrenamiento como en ejecución.
- El peso de cada una de las características determina la importancia que tiene en la decisión final.

Desventajas

- Imposibilidad de resolver directamente problemas no lineales.
- Dependencia que muestra en las características

Modelo Exponencial

Ventajas

- No requiere de gran volumen de datos históricos.
- Conseguir darle más importancia a la demanda más reciente o a la antigua.

Desventajas

- No se puede modelar relaciones complejas.
- No se pueden capturar relaciones no lineales sin transformar la entrada.

Modelo Polinomial

Ventajas

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

Desventajas

- Requiere aplicar escalamiento de dimensiones.
- Más complicado de comprender.

Cual es el principal problema del modelo probabilistico

Que este modelo puede tomar un conjunto de datos obtenidos de muestreos de datos con comportamiento que se predice aleatorio. Incluye un conjunto de asunciones sobre la generación de algunos datos muestrales de manera que se asemejen a los datos de una población mayor.

