UNIVERSIDAD POLITÉCNICA SALESIANA

EI VECINO - CUENCA

Estudiante: Gustavo Guallpa

Profesor: Ing. Diego Quisi

Asignatura: Simulación

Fecha: 23/11/2020.

Tema: Prueba Regresión.

In [143]:
```
# 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,r2_score
from scipy.optimize import curve_fit
from scipy.optimize import fsolve
from sklearn import linear_model
import matplotlib.pyplot as plt
from sklearn.preprocessing import PolynomialFeatures
%matplotlib inline
```

In [144]:
```
path='owid-covid-data.csv'
df=pd.read_csv(path)
#df.head(300)
df
```

Out[144]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	t
0	ABW	North America	Aruba	2020-03-13	2.0	2.0	NaN	
1	ABW	North America	Aruba	2020-03-19	NaN	NaN	0.286	
2	ABW	North America	Aruba	2020-03-20	4.0	2.0	0.286	
3	ABW	North America	Aruba	2020-03-21	NaN	NaN	0.286	
4	ABW	North America	Aruba	2020-03-22	NaN	NaN	0.286	
...	...	...	...	...	...	...	...	
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 [145]:
```
df = df[df['location'].isin(['Venezuela'])] #Filtro la Informacion solo para M
exico.
df = df.loc[:,['date','total_cases']] #Selecciono las columnas de analasis
FMT = '%Y-%m-%d'
date = df['date']
df['date'] = date.map(lambda x : (datetime.strptime(x, FMT) - datetime.strptim
e("2020-01-14", FMT)).days)
df
```

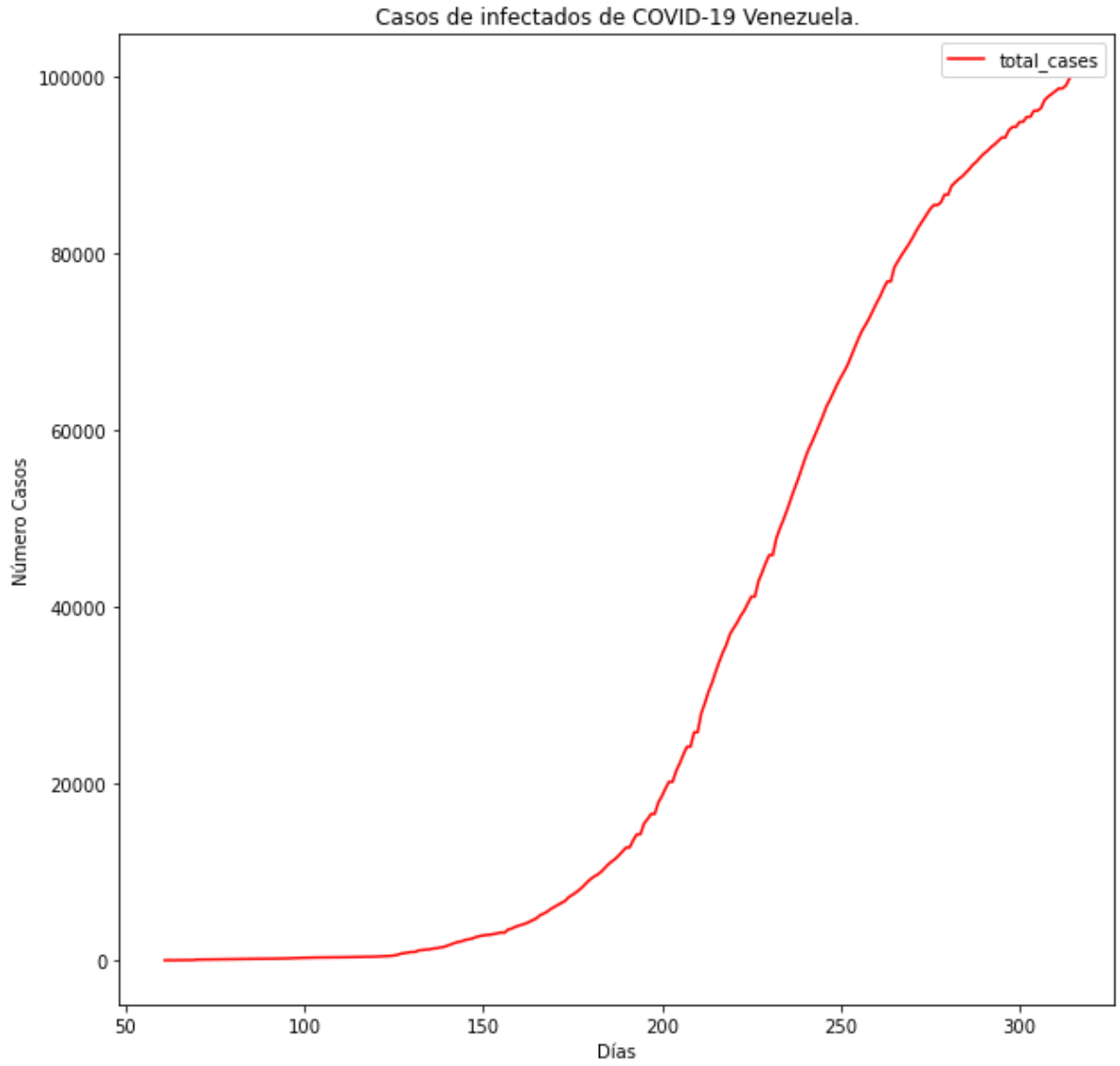
Out[145]:

	date	total_cases
55647	61	10.0
55648	62	15.0
55649	63	33.0
55650	64	33.0
55651	65	33.0
...	...	...
55896	310	98350.0
55897	311	98665.0
55898	312	98665.0
55899	313	99017.0
55900	314	99835.0

254 rows × 2 columns

In [146]:
```
plt.rcParams['figure.figsize'] = [10, 10]
# Represento los datos generados
ax = df.plot(x ='date', y='total_cases',color='red',title='Casos de infectados
de COVID-19 Venezuela.')
ax.set_xlabel("Días")
ax.set_ylabel("Número Casos")
```

Out[146]: Text(0, 0.5, 'Número Casos')

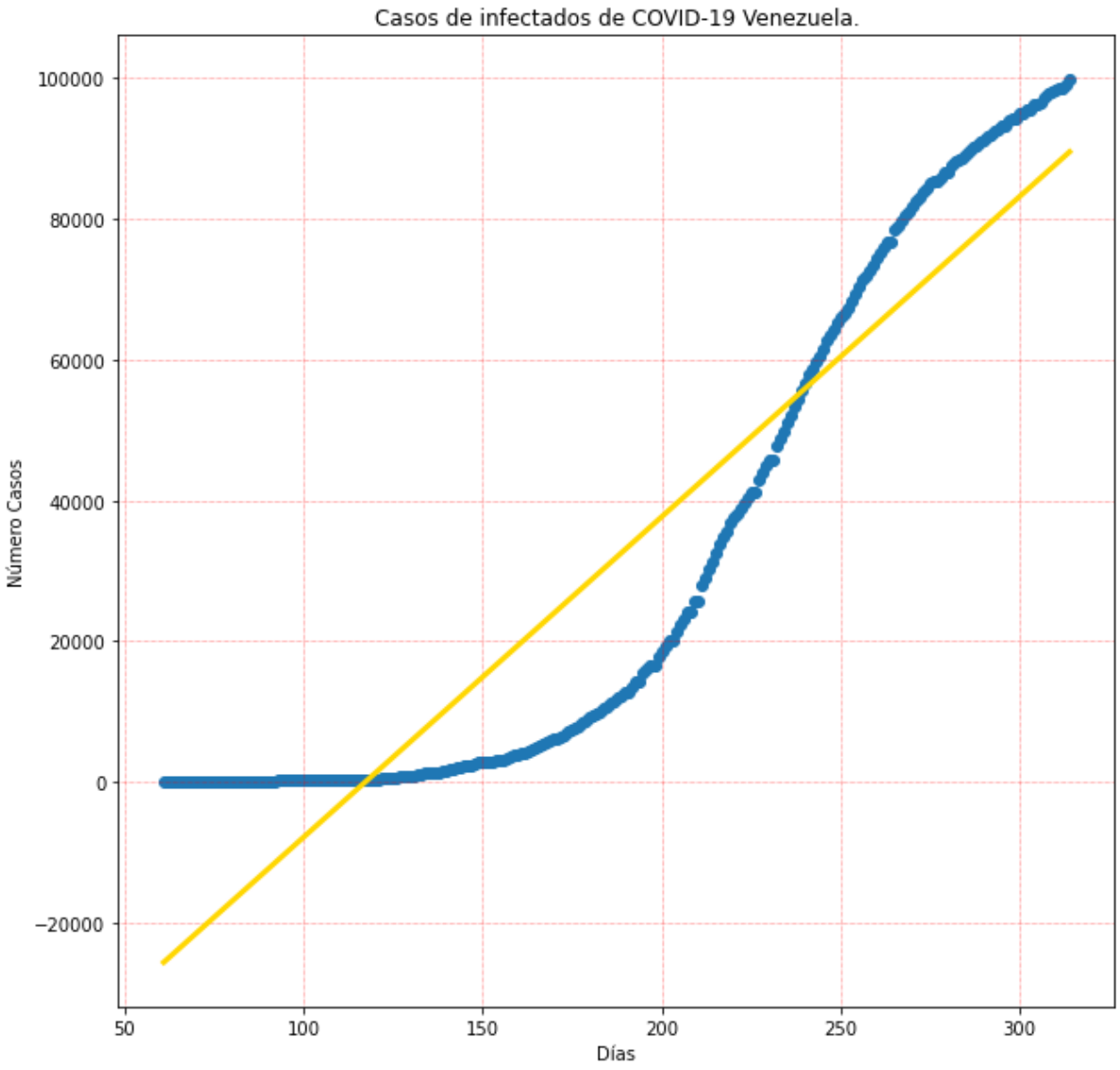


REGRESIÓN LINEAL SIMPLE

In [217]:
```
x = list(df.iloc [:, 0]) # Fecha
y = list(df.iloc [:, 1]) # Numero de casos

# Creo un modelo de regresión lineal
modelo = linear_model.LinearRegression()

# Entreno el modelo con los datos (X,Y)
modelo.fit(np.array(x).reshape(-1,1), y)

#Ahora vamoa a calcular b_0
print (u'Ordenada al origen: ', modelo.intercept_)

# Ahora puedo obtener el coeficiente b_1
print (u'Pendiente: ', modelo.coef_[0])
beta = modelo.coef_[0]#Modificar el valor de la pendiente.
# Podemos predecir usando el modelo
y_pred = modelo.predict(np.array(x).reshape(-1,1))

# Por último, calculamos el error cuadrático medio y el estadístico R^2(Precis
ión del modelo.)
print (u'Error cuadrático medio: %.2f' % mean_squared_error(y, y_pred))
print (u'Estadístico R_2: %.2f' % r2_score(y, y_pred))
```

```
Ordenada al origen:  -53339.96373668825
Pendiente:  454.93925016522445
Error cuadrático medio: 175372236.62
Estadístico R_2: 0.86
```

In [218]:
```
#Graficar
# Representamos el ajuste (rojo) y la recta Y = beta*x (verde)
plt.scatter(x, y)#Dibujo mis puntos originales
plt.plot(x, y_pred, color='gold',linewidth=3.0) #Dibujo con os valores ya pred
ecidos.

#Propiedades
plt.title('Casos de infectados de COVID-19 Venezuela.')
plt.xlabel('Días')
plt.ylabel('Número Casos')
plt.grid(color='r', linestyle='dotted', linewidth=0.5)
plt.show()
```



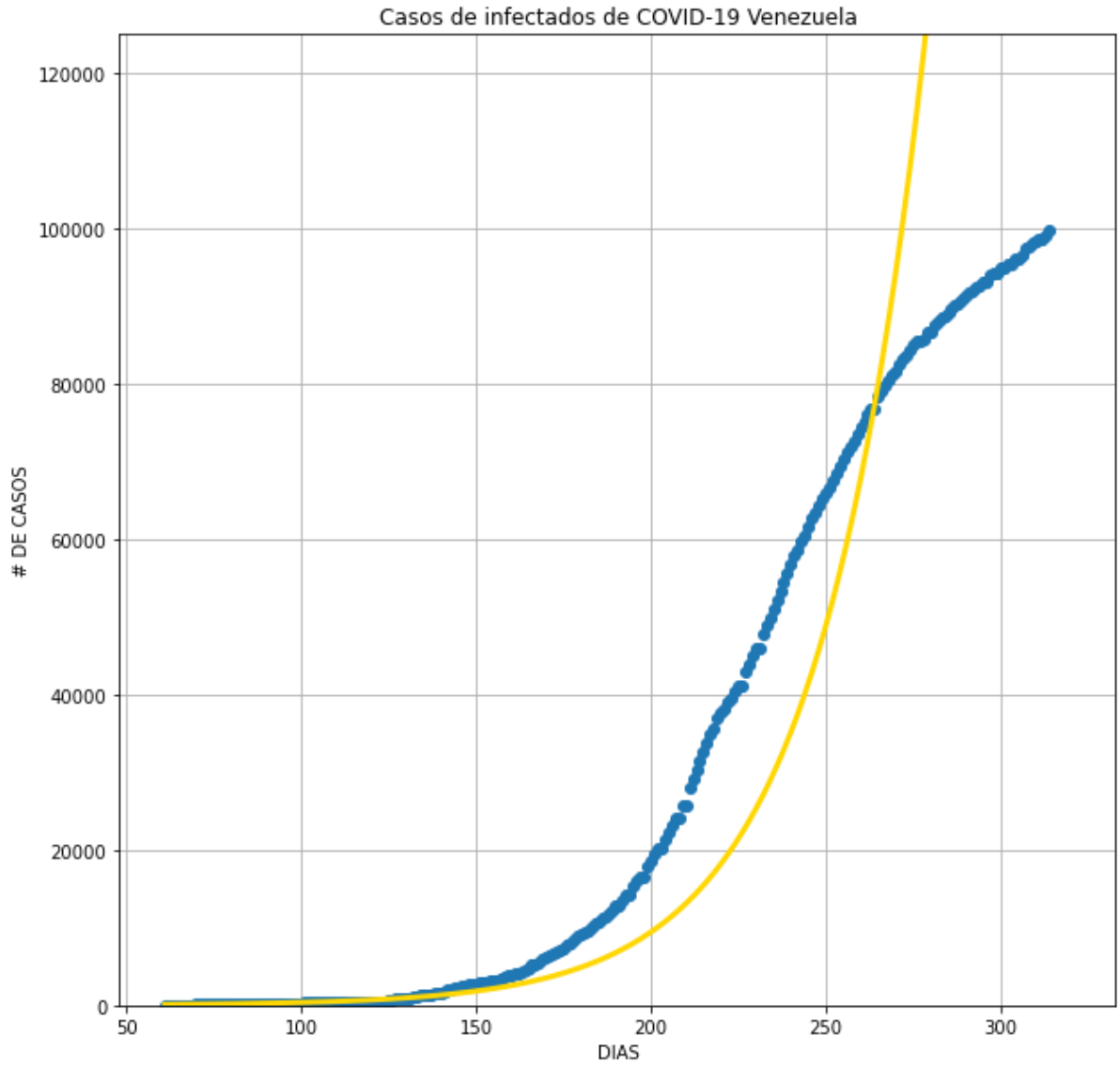
In [219]:
```
##y=mx+c
print('LA ECUACIÓN DEL MODELO ES LA SIGUIENTE:')
print('y = ', modelo.coef_, 'x ', modelo.intercept_)
#Predecir 7 días.
print(len(x))
y_semana = modelo.predict([[len(x)+7]])
print('PREDICCION PROXIMA SEMANA',y_semana)
```

```
LA ECUACIÓN DEL MODELO ES LA SIGUIENTE:
y =  [454.93925017] x  -53339.96373668825
254
PREDICCION PROXIMA SEMANA [65399.18055644]
```

REGRESION EXPONENCIAL

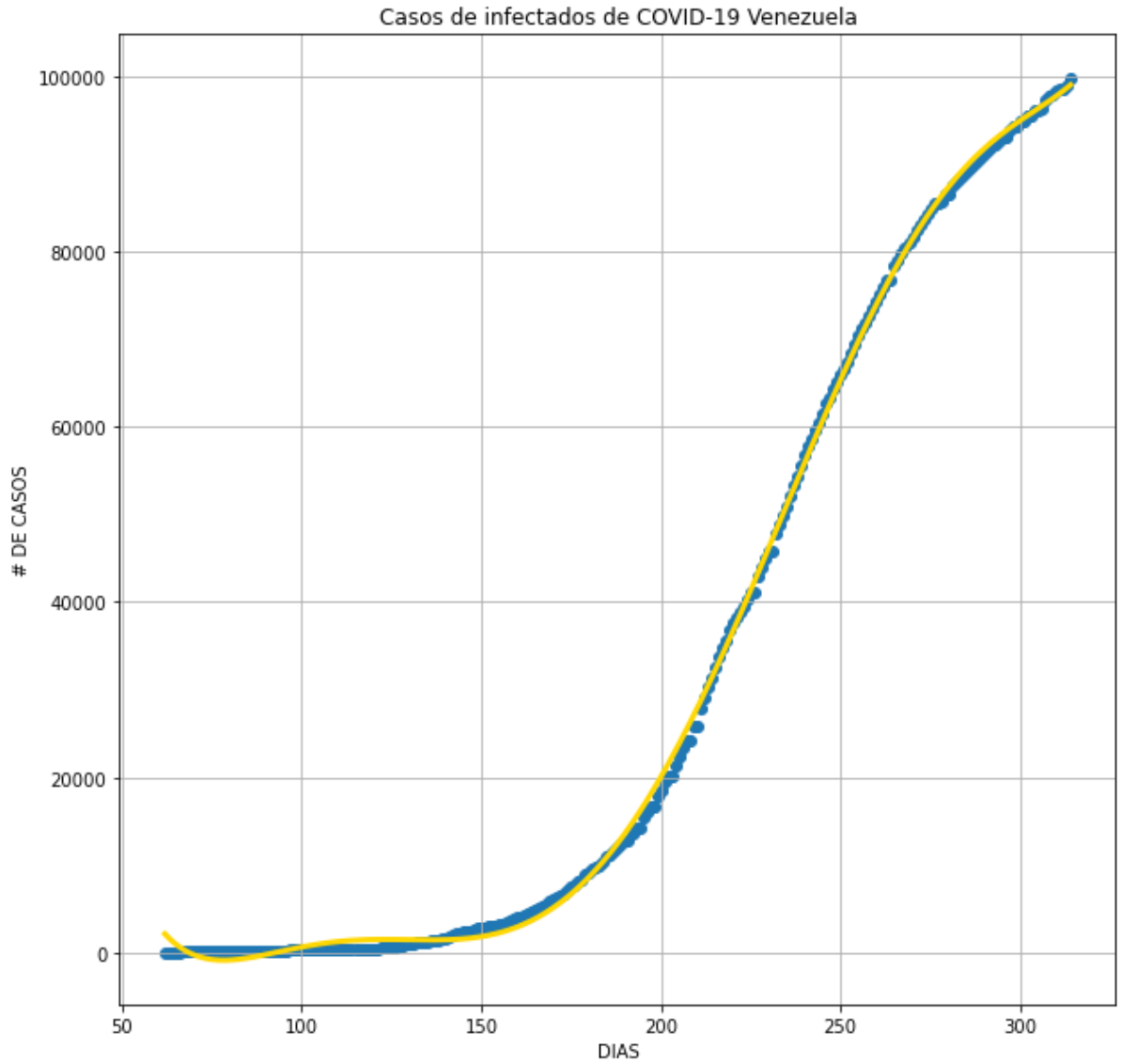
In [222]:
```
curve_fit=np.polyfit(x,np.log(y),deg=1)
print(curve_fit)
pred_x=np.array(list(range(min(x),max(x)+7)))
yx=np.exp(curve_fit[1])*np.exp(curve_fit[0]*pred_x)
plt.title('Casos de infectados de COVID-19 Venezuela')
plt.plot(x,y,"o")
plt.plot(pred_x,yx,color='gold',linewidth=3.0)
plt.ylim(-10, 125000)
plt.xlabel('DIAS')
plt.ylabel('# DE CASOS')
plt.grid(True)

print('PREDICCION PROXIMA SEMANA',yx[len(yx)-1])
```

```
[0.03292173 2.56523954]
PREDICCION PROXIMA SEMANA 489036.8399176398
```



REGRESION POLINOMICA

In [250]:
```
x = list(df.iloc [1::, 0]) # Fecha
y = list(df.iloc [1::, 1]) # Numero de casos
#Convertimos a un arreglo de 1 solo columna

x=np.array(x).reshape(-1,1)
y=np.array(y).reshape(-1,1)

#Como se veia en el caso de la regresion lineal , para obtener mejores resulta
dos lo ideal sería una curva

#Esta transformación la podemos realizar fácilmente son SkLearn gracias Polyno
mialFeatures(). A la que le indicamos el grado de la función que queremos obte
ner. Haciendo uso del método fit_transform() obtendremos el término cuadrático
que andamos buscando para nuestro modelo.
poly = PolynomialFeatures(degree=6, include_bias=False)
x_poly = poly.fit_transform(x)
#print(x)

#Defino el algoritmo a utilizar
model = linear_model.LinearRegression()

#Entrenamos
model.fit(x_poly, y)
y_pred = model.predict(x_poly)


plt.title('Casos de infectados de COVID-19 Venezuela')
plt.scatter(x, y)
plt.plot(x, y_pred, color='gold',lw=3.0)
plt.grid(True)
plt.xlabel('DIAS')
plt.ylabel('# DE CASOS')
plt.show()


#PREDICCION SIGUIENTE SEMANA
model.fit(x_poly, y)
y_pred = model.predict(x_poly)
print('PREDICCION PROXIMA SEMANA',y_pred[len(y_pred)-1])
```



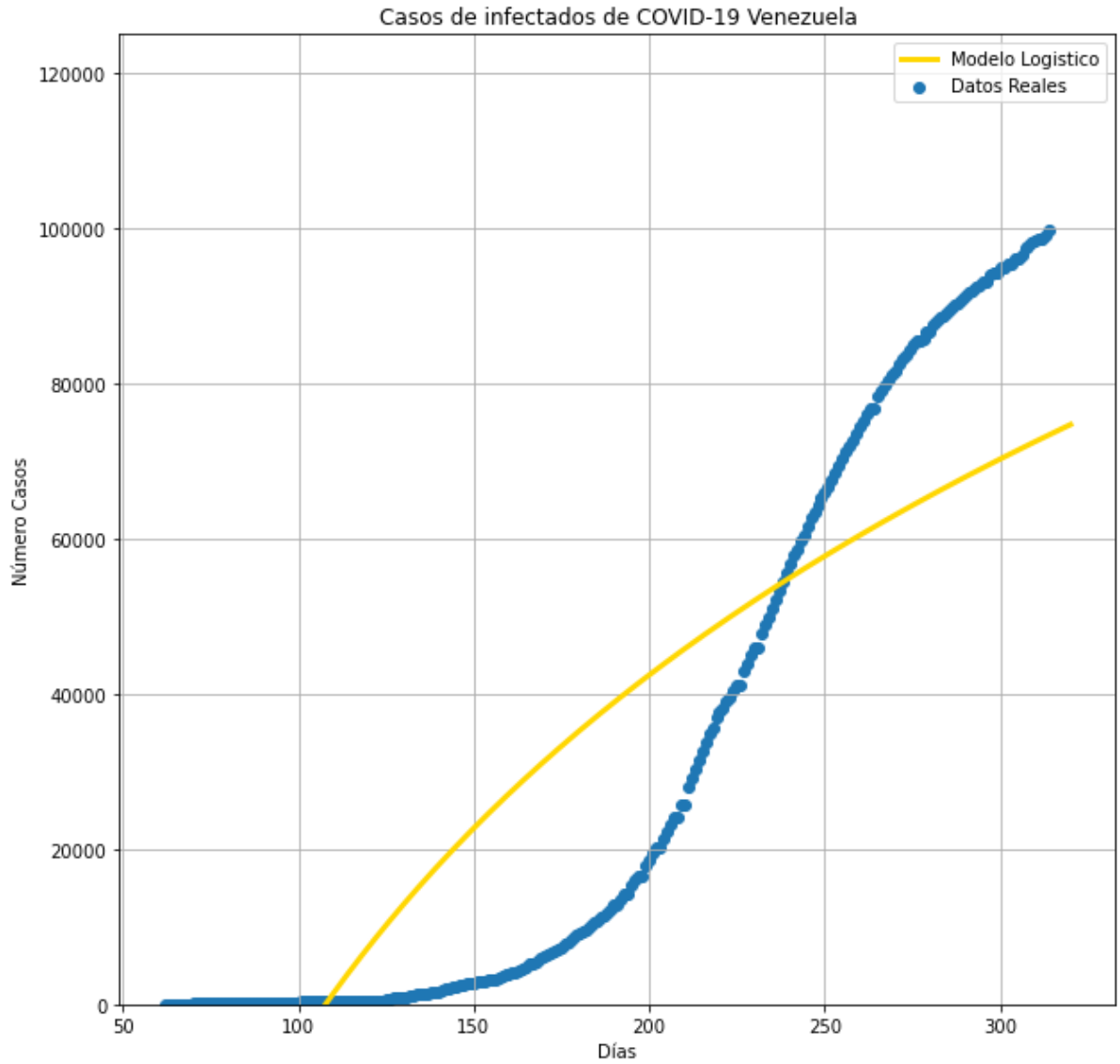

```
PREDICCION PROXIMA SEMANA [99061.06249104]
```

In [249]:
```
rmse = np.sqrt(mean_squared_error(y,y_pred))
r2 = r2_score(y,y_pred)
print ('Raíz del Error Cuadrático Medio (RMSE) : ' + str(rmse))#Mide la cantid
ad de error que hay entre dos conjuntos de datos.
print ('Coeficiente de Determinación R2: ' + str(r2))#Mientras mas cerccano a
 1 , mejor ajuste.
```

```
Raíz del Error Cuadrático Medio (RMSE) : 1534.974492263319
Coeficiente de Determinación R2: 0.9981723019473473
```

REGRESION LOGARITMICA - LOGISTICA

In [247]:
```
from scipy.optimize import curve_fit

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


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

exp_fit = curve_fit(modelo_logistico,x,y) #Extraemos los valores de los parame
tros

#print(exp_fit)

pred_x = list(range(min(x),max(x)+7)) # Predecir 7 días más
# Real data
plt.scatter(x,y,label="Datos Reales")
# Predicted exponential curve
y_res=[modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i in pred_x]
plt.plot(pred_x, [modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i in pre
d_x], label="Modelo Logistico" ,color='gold',lw=3.0)
plt.legend()
plt.title('Casos de infectados de COVID-19 Venezuela')
plt.xlabel("Días")
plt.ylabel("Número Casos")
#plt.ylim((min(y)*0.9,max(y)*3.1)) # Definir los limites de Y
plt.ylim(-10, 125000)
plt.grid(True)
plt.show()

print('PREDICCION PROXIMA SEMANA',y_res[len(pred_x)-1])
```




```
PREDICCION PROXIMA SEMANA 74677.85185748123
```

MODELO PROBABILISTICO

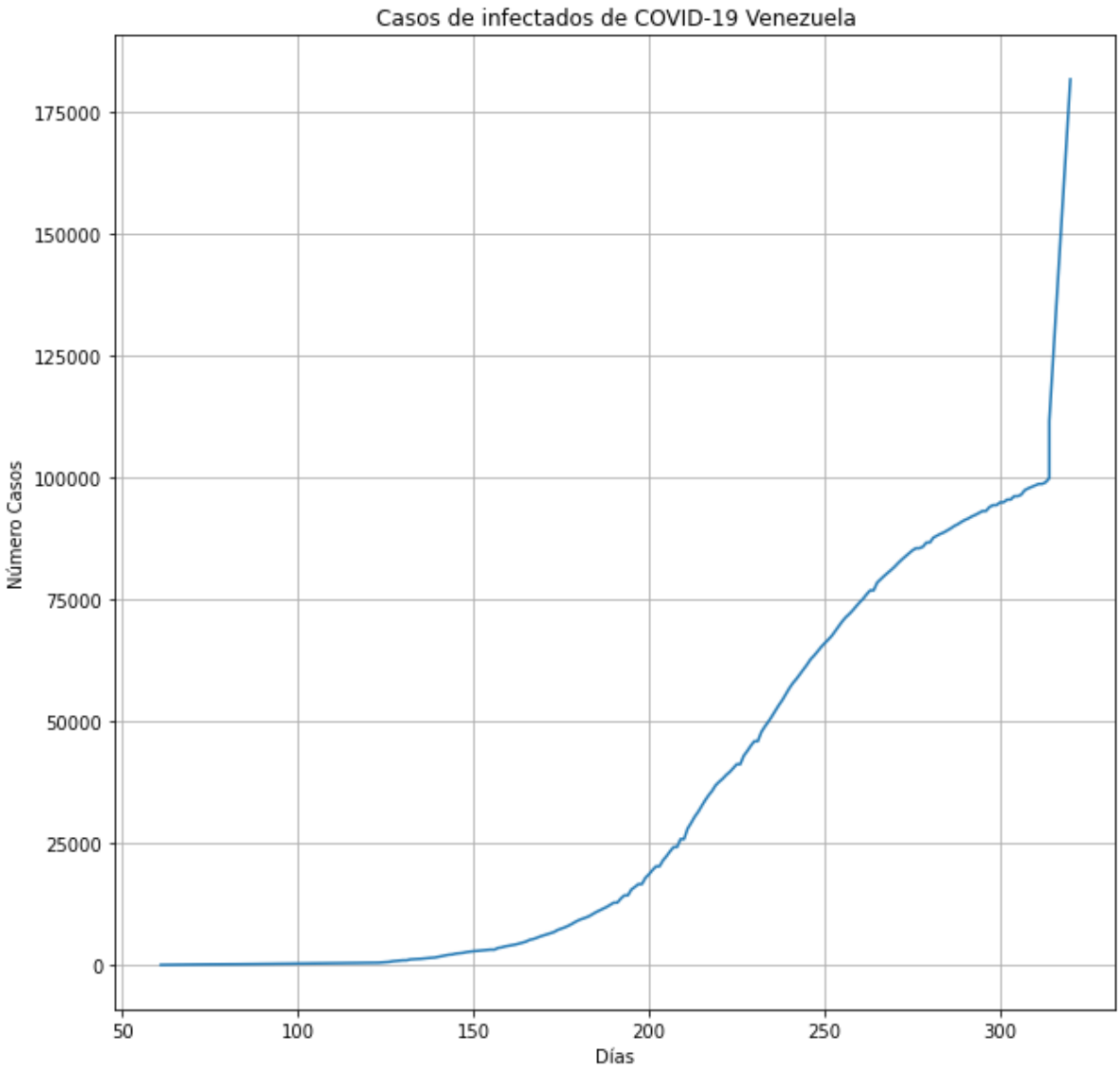
In [225]:
```
filtro = df ["total_cases"]##Filtro los datos.
#Obtenemos la mediana
media = filtro.mean()
mediana = filtro.median()
print(mediana)#Valor medio
print(media)#promedio
```

```
11687.0
31961.145669291338
```

In [245]:
```
y = list(df.iloc [:, 1]) # Total casos
x = list(df.iloc [:, 0]) # Dias
#Realizamos un ejemplo de prediccion
prediccion_siguiente = int(y[-1] + mediana)
print(prediccion_siguiente)
```

```
111522
```

In [246]:
```
# Quiero predecir cuántos "Casos" voy a obtener de aqui a 7 dias.
for i in range(x[-1], x[-1]+7):
    x.append(i)
    y.append(int(y[-1] + mediana))
valorProbabilistico=int(y[-1] + mediana)
plt.plot(x, y)
plt.title('Casos de infectados de COVID-19 Venezuela')
plt.xlabel("Días")
plt.ylabel("Número Casos")
plt.grid(True)
plt.show()
print('PREDICCION PROXIMA SEMANA',valorProbabilistico)
```




```
PREDICCION PROXIMA SEMANA 193331
```

¿Cuál tiene una mejor predicción?

Mediante el empleo de la regresión Polinomial se puede evidenciar que se puede obtener un mejor ajuste para nuestros datos y eso se corrobora con el nivel de ajuste que nos ofrece que es del 0.99% es un valor muy bueno, que nos da una idea lo que el modelo de regresion con ese ajuste puede relizar.

¿Ventajas y desventajas de los modelos?

- Modelo Lineal.

  VENTAJAS
  - Es muy fácil de modelar.
  - Modelo muy utilizado para predecir probabilidades.

  DESVENTAJAS
  - No permite modelar relaciones complejas.

- Modelo Exponencial.

  VENTAJAS
  - La regresión exponencial aunque no es lineal es linealizable tomando logaritmos ya que haciendo el cambio de variable

    v = log.

  DESVENTAJAS
  - Tiende a tener un crecimiento muy drastico a medida que aumentan los valores.

- Modelo Polinomial.

  VENTAJAS
  - Muy similar a la regresión lineal, con una ligera forma en que se trata los espacios.

  DESVENTAJAS
  - Los polinomios superiores pueden terminar produciendo resultados extraños en la extrapolación.
  - Tiende a ajustarse drásticamente incluso en un simple conjuto de datos.
- Modelo Logarítmico.

  VENTAJAS
  - Ampliamente utilizado para describir el crecimiento de una población.
  - Modelo muy bueno para clasificar.

  DESVENTAJAS
  - No es muy bueno para predecir.

¿Cuál es el principal problema del modelo probabilístico?

Dentro del modelo se puede evidenciar que es un modelo ideal para trabajar con datos que no varían mucho ya que como sabemos en este modelo lo que se hace es sacar la media(n promedio) y la mediana(valor medio). Pongamos un ejemplo digamos que si se tiene datos de COVID-19 que varian de una forma muy grande y se puede llegar a cometer un error que puede ser muy elevado, este al ser elevado no nos sirve para poder hacer una predicción correcta.

In [ ]: