

```
In [1]: # Importamos Las Librerias
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

import pandas as pd
```

```
In [2]: dataset = pd.read_csv('vacunas2.csv')

# Agregamos un fila de total casos nuevos
def obtener_total_casos(valores):
    nuevos_casos = []
    for i, valor in enumerate(valores):
        if i > 0:
            nuevo_valor = nuevos_casos[i-1] + valor
        else:
            nuevo_valor = valor
        nuevos_casos.append(nuevo_valor)
    # retornamos la nueva lista
    return nuevos_casos

dataset['Total Casos'] = obtener_total_casos(dataset['dosis_total'])
dataset
```

```
Out[2]:
```

	fecha	dosis_total	primera_dosis	segunda_dosis	Total Casos
--	-------	-------------	---------------	---------------	-------------

0	21-01-2021	0	0	0	0
1	22-01-2021	108	108	0	108
2	27-01-2021	2982	2982	0	3090
3	04-02-2021	6228	6228	0	9318
4	17-02-2021	8190	6228	1962	17508
5	24-02-2021	24492	20784	3708	42000
6	01-03-2021	42114	35886	6228	84114
7	04-03-2021	59316	53088	6228	143430
8	05-03-2021	71148	64920	6228	214578
9	08-03-2021	74472	68244	6228	289050
10	09-03-2021	75258	69030	6228	364308
11	11-03-2021	95915	89687	6228	460223
12	12-03-2021	123176	116948	6228	583399
13	13-03-2021	139359	119222	20137	722758
14	15-03-2021	141191	121054	20137	863949
15	21-03-2021	178970	140765	38205	1042919
16	23-03-2021	182261	143614	38647	1225180
17	24-03-2021	191179	152526	38653	1416359
18	26-03-2021	230770	172413	58357	1647129
19	27-03-2021	235000	174642	60358	1882129
20	29-03-2021	244866	182329	62537	2126995
21	01-04-2021	283106	204902	78204	2410101
22	04-04-2021	301069	211720	89349	2711170

	fecha	dosis_total	primera_dosis	segunda_dosis	Total Casos
23	05-04-2021	335093	228504	106589	3046263
24	06-04-2021	356783	244159	112624	3403046
25	08-04-2021	363255	250631	112624	3766301
26	14-04-2021	480962	338180	142782	4247263
27	15-04-2021	485132	338180	146952	4732395
28	16-04-2021	514151	354019	160132	5246546
29	17-04-2021	545132	377199	167933	5791678
30	18-04-2021	554369	384093	170276	6346047
31	19-04-2021	577711	401871	175840	6923758
32	20-04-2021	601229	421937	179292	7524987
33	21-04-2021	643702	457403	186299	8168689
34	22-04-2021	675510	486524	188986	8844199
35	23-04-2021	711204	514854	196350	9555403
36	24-04-2021	732717	532367	200350	10288120
37	25-04-2021	743937	541420	202517	11032057
38	26-04-2021	765489	555265	210224	11797546
39	27-04-2021	816175	595699	220476	12613721
40	28-04-2021	861393	633421	227972	13475114
41	29-04-2021	920865	691000	229865	14395979
42	30-04-2021	987452	748021	239431	15383431
43	01-05-2021	1036794	791822	244972	16420225
44	02-05-2021	1067472	821960	245512	17487697
45	04-05-2021	1141262	889218	252044	18628959
46	05-05-2021	1182085	924539	257546	19811044
47	06-05-2021	1215676	953238	262438	21026720
48	07-05-2021	1245822	981620	264202	22272542

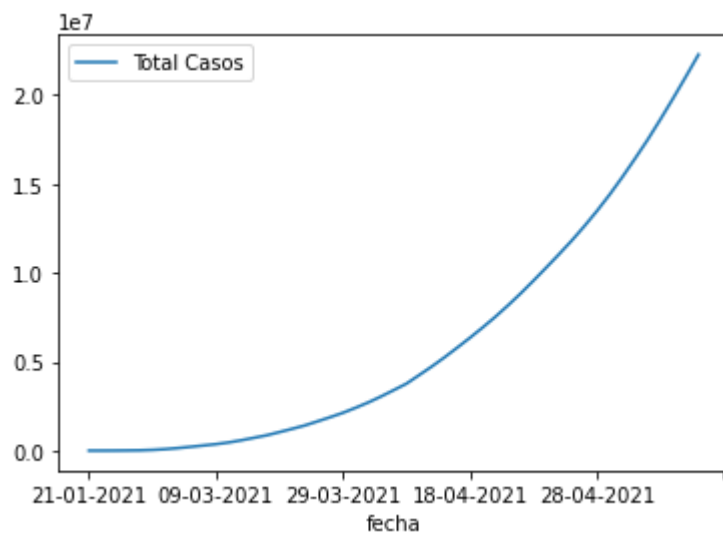
In [3]:

```
# Graficamos el dataset

dataset.plot(x='fecha', y='Total Casos')
```

Out[3]:

```
<AxesSubplot:xlabel='fecha'>
```



```
In [4]: # Convertimos la fecha alfanumerica a numerica
from datetime import datetime

FMT = '%d-%m-%Y'
dates = dataset['fecha']
dataset['Dia'] = abs(dates.map(lambda x : (datetime.strptime(x , FMT) - datetime(2021,5, 9)).days))
# Mostramos el dataset final

dataset
```

```
Out[4]:
```

	fecha	dosis_total	primera_dosis	segunda_dosis	Total Casos	Dia
0	21-01-2021	0	0	0	0	108
1	22-01-2021	108	108	0	108	107
2	27-01-2021	2982	2982	0	3090	102
3	04-02-2021	6228	6228	0	9318	94
4	17-02-2021	8190	6228	1962	17508	81
5	24-02-2021	24492	20784	3708	42000	74
6	01-03-2021	42114	35886	6228	84114	69
7	04-03-2021	59316	53088	6228	143430	66
8	05-03-2021	71148	64920	6228	214578	65
9	08-03-2021	74472	68244	6228	289050	62
10	09-03-2021	75258	69030	6228	364308	61
11	11-03-2021	95915	89687	6228	460223	59
12	12-03-2021	123176	116948	6228	583399	58
13	13-03-2021	139359	119222	20137	722758	57
14	15-03-2021	141191	121054	20137	863949	55
15	21-03-2021	178970	140765	38205	1042919	49
16	23-03-2021	182261	143614	38647	1225180	47
17	24-03-2021	191179	152526	38653	1416359	46
18	26-03-2021	230770	172413	58357	1647129	44
19	27-03-2021	235000	174642	60358	1882129	43
20	29-03-2021	244866	182329	62537	2126995	41
21	01-04-2021	283106	204902	78204	2410101	38
22	04-04-2021	301069	211720	89349	2711170	35

	fecha	dosis_total	primera_dosis	segunda_dosis	Total Casos	Dia
23	05-04-2021	335093	228504	106589	3046263	34
24	06-04-2021	356783	244159	112624	3403046	33
25	08-04-2021	363255	250631	112624	3766301	31
26	14-04-2021	480962	338180	142782	4247263	25
27	15-04-2021	485132	338180	146952	4732395	24
28	16-04-2021	514151	354019	160132	5246546	23
29	17-04-2021	545132	377199	167933	5791678	22
30	18-04-2021	554369	384093	170276	6346047	21
31	19-04-2021	577711	401871	175840	6923758	20
32	20-04-2021	601229	421937	179292	7524987	19
33	21-04-2021	643702	457403	186299	8168689	18
34	22-04-2021	675510	486524	188986	8844199	17
35	23-04-2021	711204	514854	196350	9555403	16
36	24-04-2021	732717	532367	200350	10288120	15
37	25-04-2021	743937	541420	202517	11032057	14
38	26-04-2021	765489	555265	210224	11797546	13
39	27-04-2021	816175	595699	220476	12613721	12
40	28-04-2021	861393	633421	227972	13475114	11
41	29-04-2021	920865	691000	229865	14395979	10
42	30-04-2021	987452	748021	239431	15383431	9
43	01-05-2021	1036794	791822	244972	16420225	8
44	02-05-2021	1067472	821960	245512	17487697	7
45	04-05-2021	1141262	889218	252044	18628959	5
46	05-05-2021	1182085	924539	257546	19811044	4
47	06-05-2021	1215676	953238	262438	21026720	3
48	07-05-2021	1245822	981620	264202	22272542	2

fabricante por fechas

```
In [5]: datos = 'https://raw.githubusercontent.com/andrab/ecuacovid/master/datos_crudos/vacunas/fabrica
datos_fabrica = pd.read_csv(datos)
datos_fabrica
```

```
Out[5]:
```

	vaccine	total	arrived_at	contract
0	Pfizer/BioNTech	8190	20/01/2021	Government of Ecuador with Pfizer
1	Pfizer/BioNTech	16380	17/02/2021	Government of Ecuador with Pfizer
2	Pfizer/BioNTech	17550	24/02/2021	Government of Ecuador with Pfizer
3	Pfizer/BioNTech	31590	03/03/2021	Government of Ecuador with Pfizer
4	Sinovac	20000	06/03/2021	Donation from the Government of Chile to the G...
5	Pfizer/BioNTech	73710	10/03/2021	Government of Ecuador with Pfizer

	vaccine	total	arrived_at	contract
6	Oxford/AstraZeneca	84000	17/03/2021	Government of Ecuador with COVAX
7	Pfizer/BioNTech	62010	17/03/2021	Government of Ecuador with Pfizer
8	Pfizer/BioNTech	65520	24/03/2021	Government of Ecuador with Pfizer
9	Pfizer/BioNTech	66690	31/03/2021	Government of Ecuador with Pfizer
10	Pfizer/BioNTech	53820	05/04/2021	Government of Ecuador with Pfizer
11	Sinovac	300000	07/04/2021	Government of Ecuador with Sinovac
12	Sinovac	700000	10/04/2021	Government of Ecuador with Sinovac
13	Pfizer/BioNTech	53820	14/04/2021	Government of Ecuador with Pfizer
14	Pfizer/BioNTech	54990	21/04/2021	Government of Ecuador with Pfizer
15	Oxford/AstraZeneca	336000	24/04/2021	Government of Ecuador with COVAX
16	Pfizer/BioNTech	54990	28/04/2021	Government of Ecuador with Pfizer
17	Pfizer/BioNTech	100620	04/05/2021	Government of Ecuador with Pfizer
18	Sinovac	3520	16/05/2021	Managed by Conmebol and donation by Sinovac
19	Pfizer/BioNTech	170820	18/05/2021	Government of Ecuador with Pfizer
20	Oxford/AstraZeneca	204000	22/05/2021	Government of Ecuador with AstraZeneca
21	Pfizer/BioNTech	170820	25/05/2021	Government of Ecuador with Pfizer
22	Sinovac	500000	29/05/2021	Government of Ecuador with Sinovac
23	Sinovac	200000	29/05/2021	Donation from the Government of China to the G...
24	Pfizer/BioNTech	107640	01/06/2021	Government of Ecuador with Pfizer

In [6]: `# generar grafica`

```

In [7]: pfizer = datos_fabrica[datos_fabrica['vaccine'].isin(['Pfizer/BioNTech'])]
sinovac = datos_fabrica[datos_fabrica['vaccine'].isin(['Sinovac'])]
oxford = datos_fabrica[datos_fabrica["vaccine"].isin(['Oxford/AstraZeneca'])]

ptot = pfizer["total"]
pdat = pfizer["arrived_at"]
stot = sinovac["total"]
sdat = sinovac["arrived_at"]
otot = oxford["total"]
odat = oxford["arrived_at"]

psum = ptot.values.sum()
ssum = stot.values.sum()
osum = otot.values.sum()

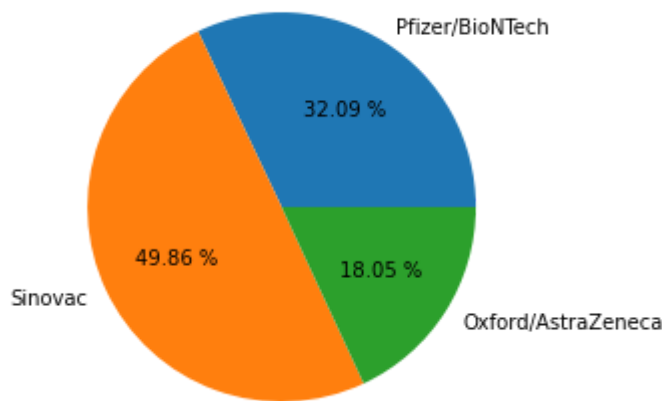
datos = [psum,ssum, osum]
facts = ['Pfizer/BioNTech', 'Sinovac', 'Oxford/AstraZeneca']

#print(datos.values)
plt.pie(datos, labels=facts, autopct="%0.2f %%")

plt.axis("equal")
plt.title("VACUNAS FABRICADAS")
plt.show()

```

VACUNAS FABRICADAS



```
In [8]: month= datos_fabrica['arrived_at']
me = []
for m in month:
    mes = m.split("/")
    me.append(int(mes[1]))
datos_fabrica["month"]=me

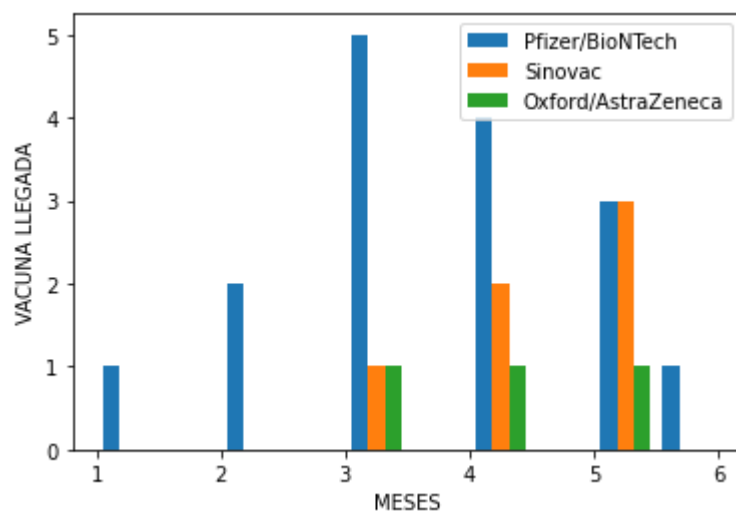
pfi = datos_fabrica[datos_fabrica['vaccine'].isin(['Pfizer/BioNTech'])]
pfi = pfi.loc[:,['total', 'month']]

sin = datos_fabrica[datos_fabrica['vaccine'].isin(['Sinovac'])]
sin = sin.loc[:,['total', 'month']]

oxf = datos_fabrica[datos_fabrica['vaccine'].isin(['Oxford/AstraZeneca'])]
oxf = oxf.loc[:,['total', 'month']]

plt.xlabel("MESES")
plt.ylabel("VACUNA LLEGADA")

plt.hist([pfi["month"], sin["month"],oxf['month']], label=['Pfizer/BioNTech', 'Sinovac', 'Oxford/AstraZeneca'], loc="upper right")
plt.show()
print(sin)
```



	total	month
4	20000	3
11	300000	4
12	700000	4
18	3520	5
22	500000	5
23	200000	5

Modelo lineal

```
In [9]: # importamos La libreria
```

```
from sklearn import linear_model
```

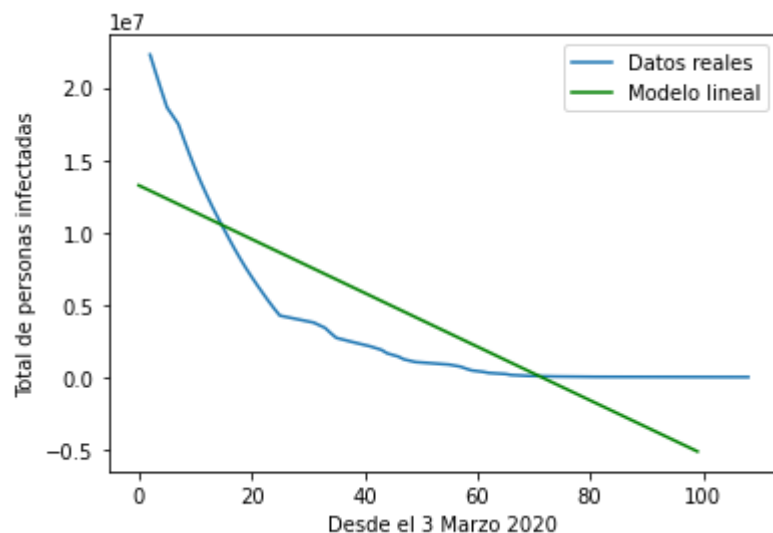
```
In [10]: x = list(dataset['Dia']) # Fecha
y = list(dataset['Total Casos']) # Numero de casos
# Creamos el objeto de Regresión Lineal
regr = linear_model.LinearRegression()

# Entrenamos nuestro modelo
regr.fit(np.array(x).reshape(-1, 1), y)

# Veamos Los coeficientes 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_)
```

Coefficients: [-185709.90507729]
Independent term: 13246266.0577565

```
In [11]: # Graficamos La funcion
plt.rc('font', size=10)
plt.plot(x, y, label="Datos reales")
x_real = np.array(range(0, 100))
plt.plot(x_real, regr.predict(x_real.reshape(-1, 1)), color='green', label="Modelo lineal")
plt.legend()
plt.xlabel("Desde el 3 Marzo 2020")
plt.ylabel("Total de personas infectadas")
plt.show()
```



```
In [12]: # Predecimos total infectados dentro de 7 dias
preducion_lineal = regr.predict([[334]])
print("El número de infectados será: ", abs(int(preducion_lineal)))
```

El número de infectados será: 48780842

MODELO EXPONENCIAL

```
In [13]: from scipy.optimize import curve_fit
x = np.array(dataset["Dia"])
y = np.array(dataset["Total Casos"])

def func(x, a, b, c, d):
    return a*x**3 + b*x**2 + c*x + d

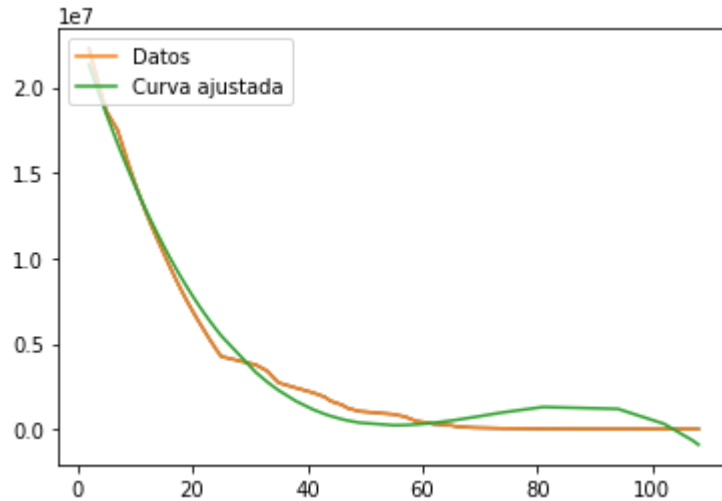
res1, res2 = curve_fit(func, x, y)
print(res1)
print(res2)
```

[-7.43414698e+01 1.58224488e+04 -1.06693361e+06 2.34112443e+07]
[[2.11627809e+01 -3.41434342e+03 1.42266306e+05 -1.22434062e+06]

```
[ -3.41434342e+03  5.65018685e+05 -2.43635715e+07  2.19218046e+08]  
[ 1.42266306e+05 -2.43635715e+07  1.11049323e+09 -1.09103233e+10]  
[ -1.22434062e+06  2.19218046e+08 -1.09103233e+10  1.35003114e+11]]
```

In [14]:

```
#Grafica  
plt.plot(x,y)  
plt.plot(x,y, label="Datos")  
  
plt.plot(x, func(x, *res1), label="Curva ajustada")  
  
plt.legend(loc='upper left')  
plt.show()
```

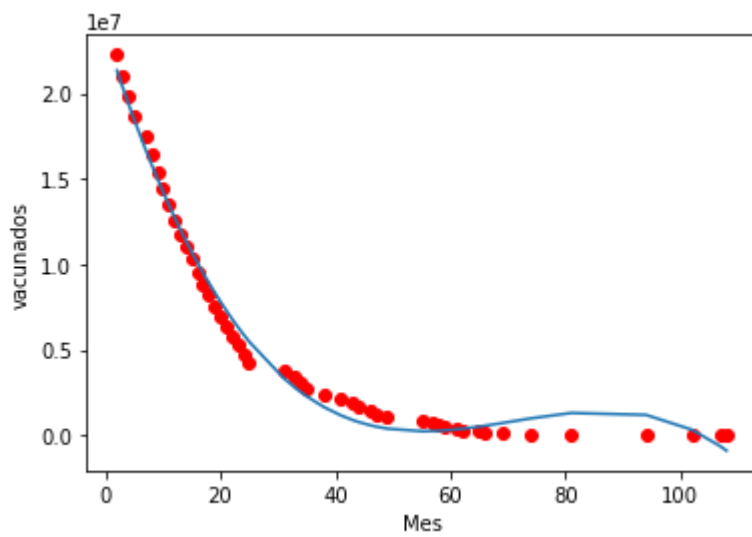


MODELO Polinomico

In [15]:

```
from sklearn.linear_model import LinearRegression  
from sklearn.preprocessing import PolynomialFeatures  
  
x = x.reshape(-1,1)  
y = y.reshape(-1,1)  
  
model = LinearRegression()  
pre_proces = PolynomialFeatures(degree=3)  
  
x_polin = pre_proces.fit_transform(x)  
  
pr_model = LinearRegression()  
pr_model.fit(x_polin,y)  
y_pred = pr_model.predict(x_polin)  
  
plt.scatter(x,y,c="red")  
plt.xlabel("Mes")  
plt.ylabel("vacunados")  
plt.plot(x,y_pred)
```

Out[15]: [



```
In [16]: y_mes = pr_model.predict(pre_proces.fit_transform([[12]]))
y_mes
```

```
Out[16]: array([[12758011.47765849]])
```

DATOS DE OTROS PAISES

```
In [17]: url_eu = 'https://raw.githubusercontent.com/MinCiencia/Datos-COVID19/master/input/Vacunacion/WC
df_eu = pd.read_csv(url_eu, sep=";", encoding="latin-1")

month= df_eu['FECHA_INMUNIZACION']
me = []
for m in month:
    mes = m.split("/")
    me.append(int(mes[1]))
df_eu["mes"]=me

tot = np.array(df_eu["SUM_of_SUM_of_1aDOSIS"])+np.array(df_eu['SUM_of_SUM_of_2aDOSIS'])
df_eu["total"]=tot
df_eu
```

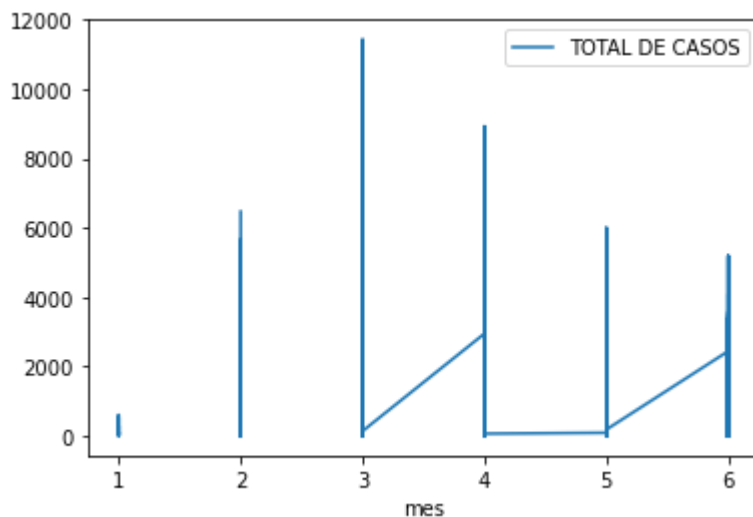
```
Out[17]:
```

	REGION_CORTO	COD_COMUNA_FINAL	FECHA_INMUNIZACION	SUM_of_SUM_of_1aDOSIS	SUM_of_SUM_of
0	Metropolitana Santiago	13101	24/12/2020	280.0	
1	Metropolitana Santiago	13108	24/12/2020	70.0	
2	Metropolitana Santiago	13123	24/12/2020	70.0	
3	Bíobío	8101	25/12/2020	406.0	
4	Bíobío	8107	25/12/2020	45.0	
...	
34367	Ñuble	16302	04/06/2021	NaN	
34368	Ñuble	16303	04/06/2021	66.0	
34369	Ñuble	16304	04/06/2021	26.0	
34370	Ñuble	16305	04/06/2021	42.0	
34371	Ñuble	16305	04/06/2021	NaN	

34372 rows × 8 columns

```
In [18]: df_eu.plot(x="mes", y="total", label="TOTAL DE CASOS")
```

```
Out[18]: <AxesSubplot:xlabel='mes'>
```



Como podemos observar en la tabla de resultados de predicciones de los modelos, concluimos que el modelo polinomico y probabilistico mantienen resultados muy aproximados a los datos actuales por lo que los consideramos los mas acertados en prediccion, sin embargo no decimos que sean los mas funcionales todo va a depender del conjunto de datos que tengamos.

Ventajas y Desventajas

Lineal Ventajas Facil de entender y explicar, lo que es una ventaja al momento de exponer frente a un publico, Es rapido de modelar y 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 recursos La prediccion mejora con datos Historicos. Desventajas No puede resolver directamente problemas no lineales La dependencia de las carateristicas es un proble es al tener datos historicos que dependan uno del otro, el modelo no podra definir otros datos que no cumplan con esta dependencia 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 prediccion. Desventajas Dependera mucho el grado de precion de como se genere dicha ecuacion exponencial, cuales son susa variables de

A =poblacion Inicial r =tasa de crecimiento t =unidades de tiempo $f(t)=A \cdot r \cdot \exp(t)$ Tambien la respuesta a la tendencia es problema ya que si day datos historicos que tenga una gran tendencia al tener otro valor que no cumpla con esta tendencia la predcion sera mas erronea

Principal problema del modelo probabilistico El problema principal es que el modelo predice de forma 'adecuada' cuando los valores del dataset son pequeños pero al momento de tener valores grandes en este conjunto de datos, su prediccion se vuelve totalmente erronea.

```
In [ ]:
```