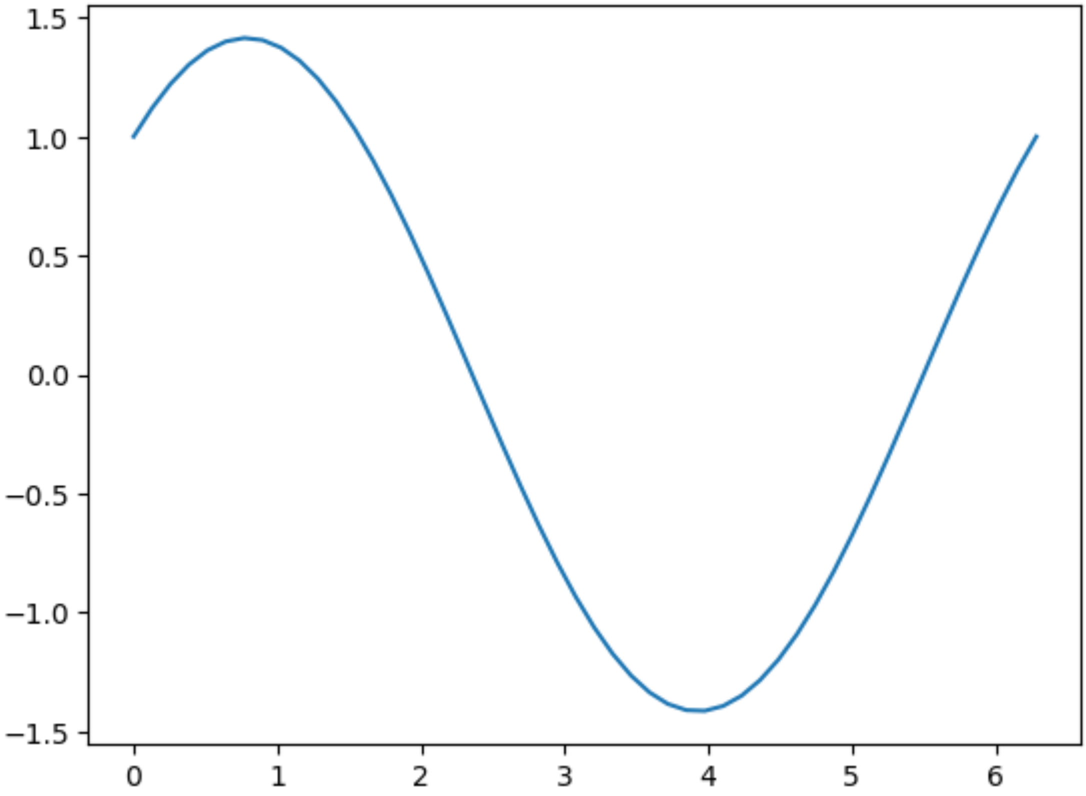
In [1]:

```
import matplotlib as plt
import pandas as pd
import numpy as np
%matplotlib inline

df = pd.read_csv("bmw.csv")
```

Ejercicio 1

In [2]:

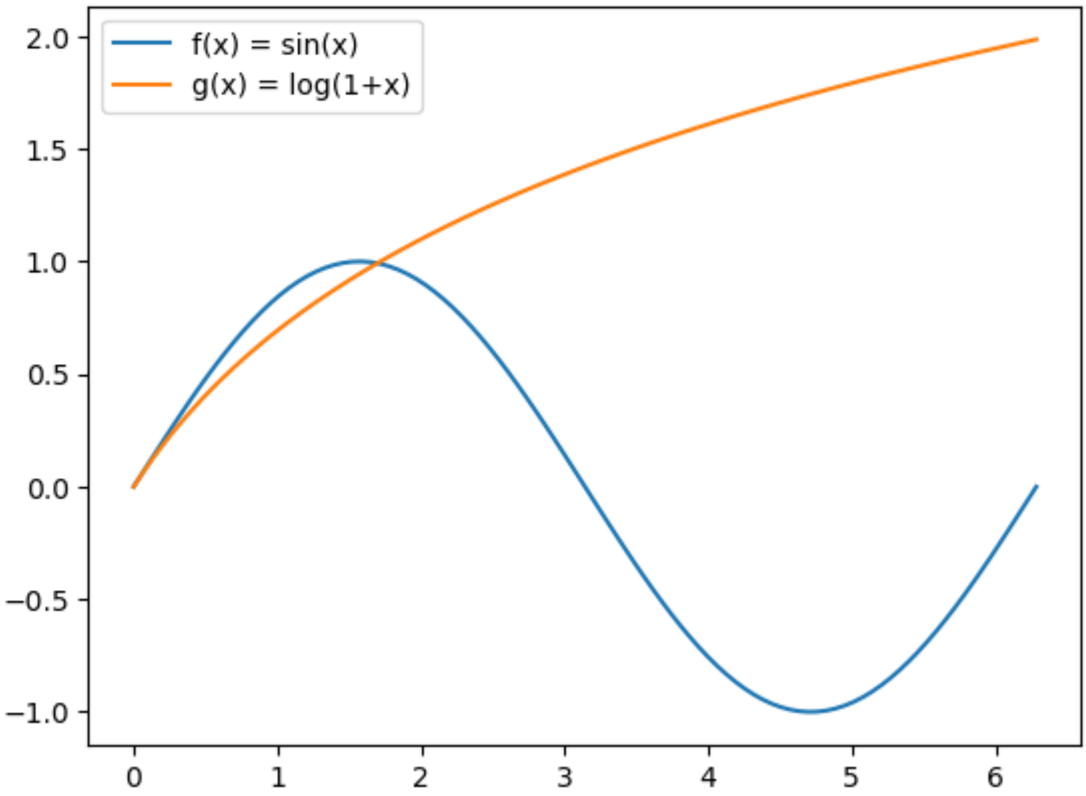
```
rango = np.linspace(0,2*np.pi) # 50 valores dfl
funcion = np.sin(rango) + np.cos(rango)
plt.pyplot.plot(rango,funcion)
plt.pyplot.show()
```



Ejercicio 2

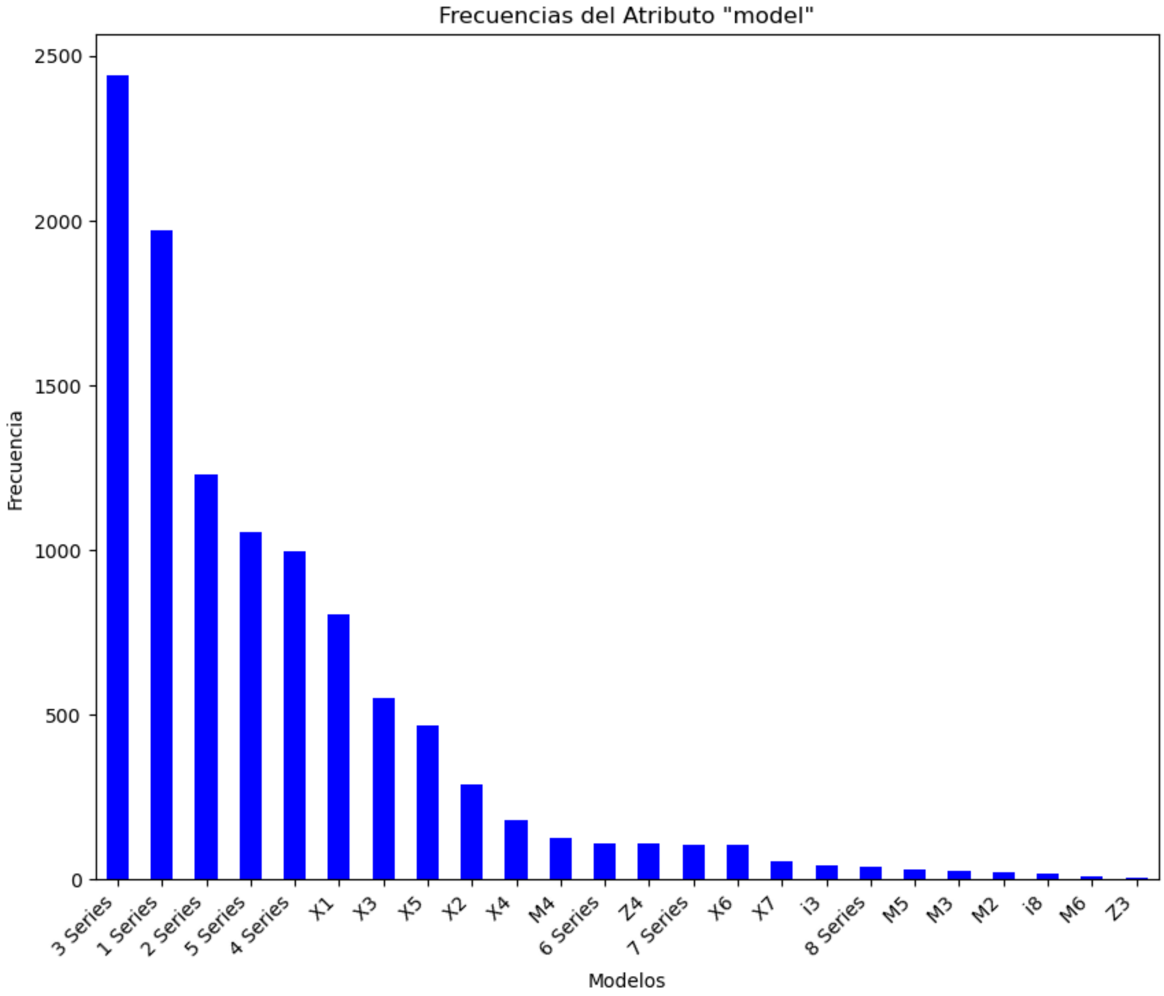
In [3]:

```
x = np.linspace(0,2*np.pi, 100)
f = np.sin(x)
g = np.log(1+x)

plt.pyplot.plot(x,f,label="f(x) = sin(x)")
plt.pyplot.plot(x,g,label="g(x) = log(1+x)")
plt.pyplot.legend()
plt.pyplot.show()
```



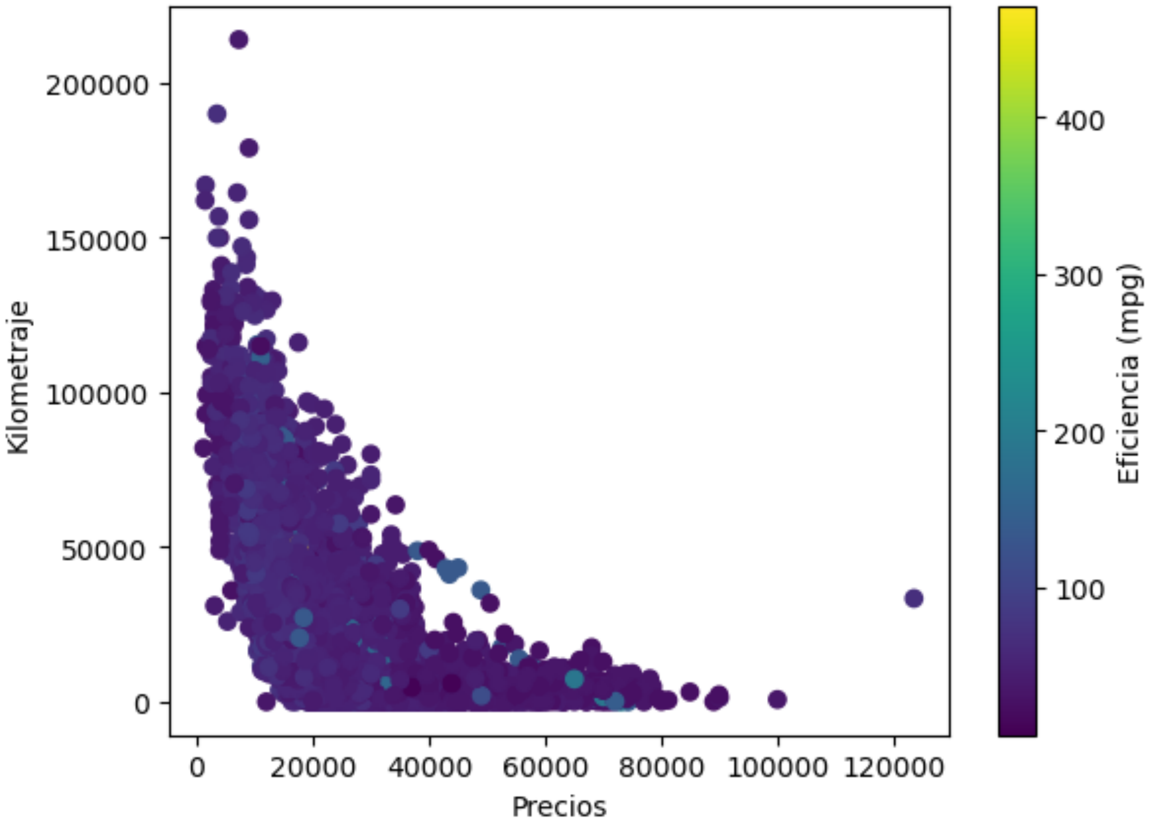
Ejercicio 3

```
In [4]: model_frequencies = df['model'].value_counts()
        plt.pyplot.figure(figsize=(10, 8))
        model_frequencies.plot(kind='bar', color='blue')

        plt.pyplot.title('Frecuencias del Atributo "model"')
        plt.pyplot.xlabel('Modelos')
        plt.pyplot.ylabel('Frecuencia')

        plt.pyplot.xticks(rotation=45, ha='right')
        plt.pyplot.show()
```



Ejercicio 4

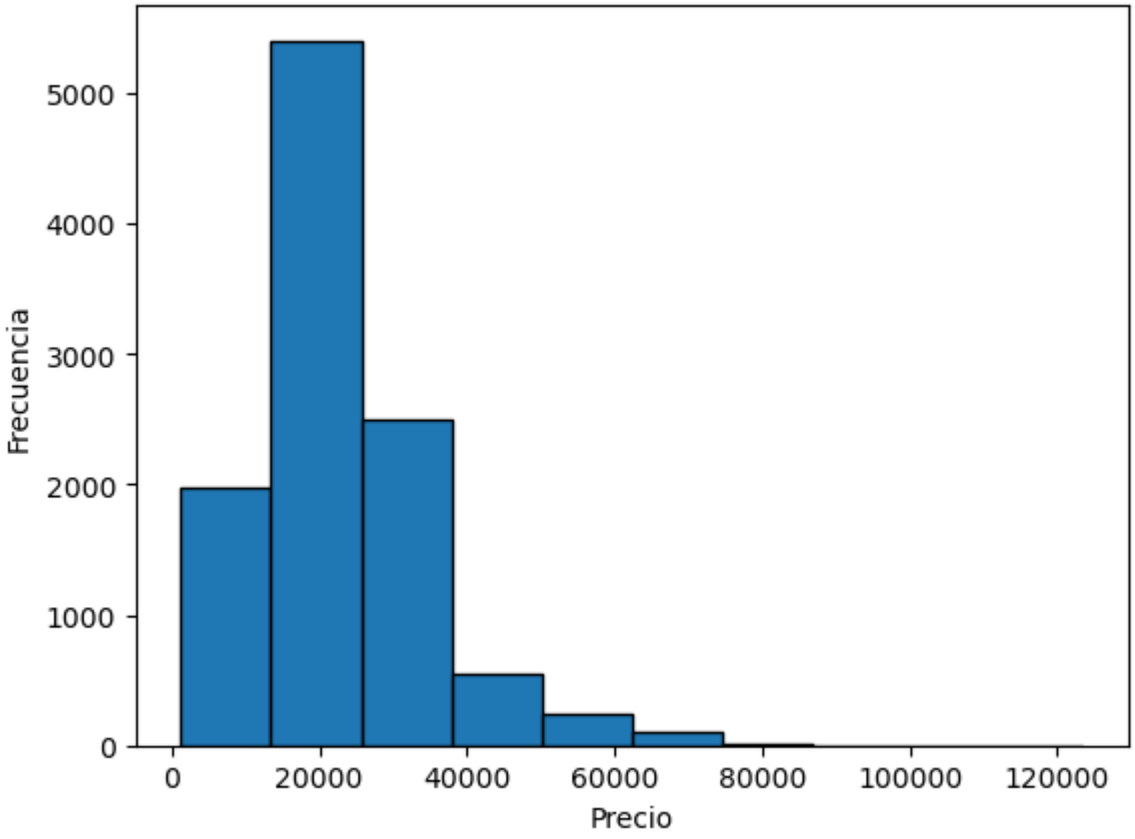
In [5]:
```
precios = df["price"]
kilometrajes = df["mileage"]
mpg = df["mpg"]
grafica = plt.pyplot.scatter(x=precios,y=kilometrajes,c=mpg)
cbar = plt.pyplot.colorbar(grafica)
cbar.set_label('Eficiencia (mpg)')
plt.pyplot.xlabel("Precios")
plt.pyplot.ylabel("Kilometraje")
plt.pyplot.show()
```


200000
150000
100000
50000
0
Kilometraje
0
20000
40000
60000
80000
100000
120000
Precios
400
300
200
100
Eficiencia (mpg)

Ejercicio 5

In [6]:

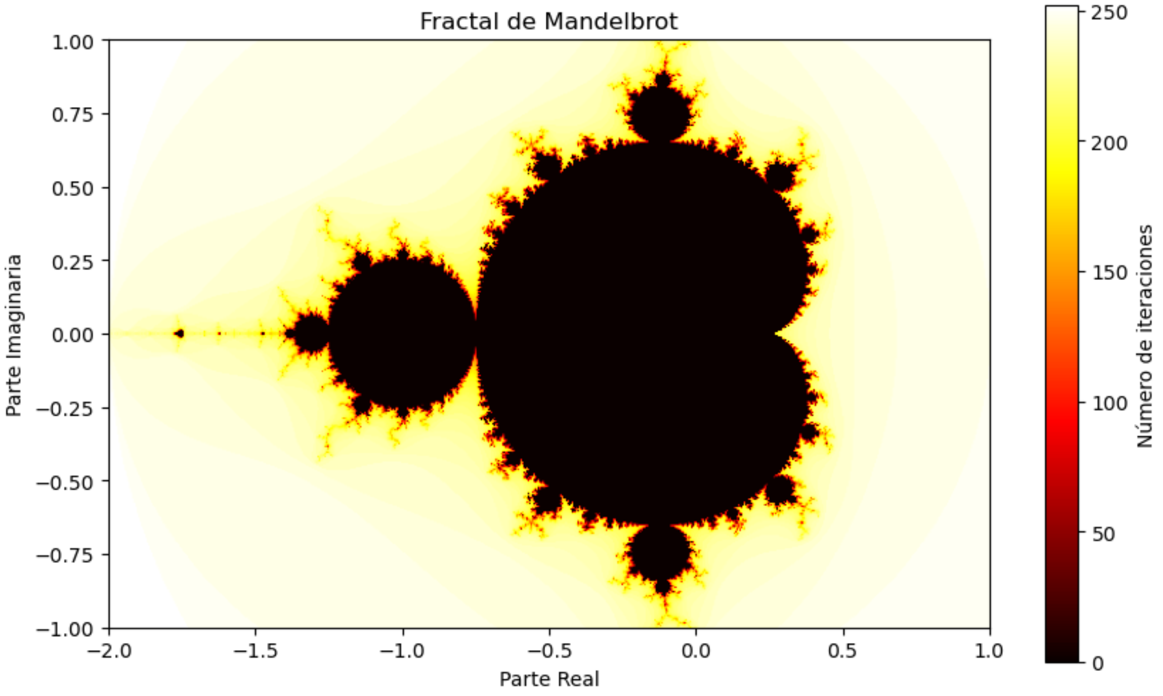
```
precios = df["price"]
plt.pyplot.hist(precios, edgecolor="black", bins=10)
plt.pyplot.xlabel("Precio")
plt.pyplot.ylabel("Frecuencia")
plt.pyplot.show()
```



Ejercicio 6

In [7]:

```
MAXIMO_ITERACIONES = 80
def mandelbrot(a,b):
    c = complex(a,b)
    z = 0
    n = 0
    while abs(z) <= 2 and n < MAXIMO_ITERACIONES:
        z = z*z + c
        n += 1
    color_pixel = 255 - int(n * 255 / MAXIMO_ITERACIONES)
    return color_pixel

ancho = 600
alto = 400
minReal, maxReal = -2.0,1.0
minImg, maxImg = -1.0,1.0

real = np.linspace(minReal,maxReal,ancho)
img = np.linspace(minImg,maxImg,alto)
real, img = np.meshgrid(real, img)
mandelbrot_set = np.vectorize(mandelbrot)(real, img) # se aplica la funci

plt.pyplot.figure(figsize=(10, 6))
plt.pyplot.imshow(mandelbrot_set, cmap='hot', extent=[minReal, maxReal, mi
plt.pyplot.colorbar(label='Número de iteraciones')
plt.pyplot.title('Fractal de Mandelbrot')
plt.pyplot.xlabel('Parte Real')
plt.pyplot.ylabel('Parte Imaginaria')
plt.pyplot.show()
```


Fractal de Mandelbrot
Parte Imaginaria
Parte Real
Número de iteraciones