

In [2]:

```
1  # GENERACIÓN DE AUDIO
2
3  # Importar Librerías
4  import numpy as np
5  import matplotlib.pyplot as plt
6  from scipy.io.wavfile import write
7
8  # Archivo de salida en el cual se grabará el audio
9  archivo_salida = 'audio_generado.wav'
10
11 # Especificar los parámetros del audio
12 duracion = 4 # in seconds
13 frecuencia_muestreo = 44100 # in Hz
14 frecuencia_tono = 784
15 valor_minimo = -4 * np.pi
16 valor_maximo = 4 * np.pi
17
18 # Generar la señal de audio
19 t = np.linspace(valor_minimo, valor_maximo, duracion * frecuencia_muestreo)
20 senial = np.sin(2 * np.pi * frecuencia_tono * t)
21
22 # Agregar algún ruido a la señal
23 ruido = 0.5 * np.random.rand(duracion * frecuencia_muestreo)
24 senial += ruido
25
26 # Escalar a valores enteros de 16 bits
27 factor_escalamiento = np.power(2, 15) - 1
28 senial_normalizada = senial / np.max(np.abs(senial))
29 senial_escalada = np.int16(senial_normalizada * factor_escalamiento)
30
31 # Almacenar la señal de audio en el archivo de salida
32 write(archivo_salida, frecuencia_muestreo, senial_escalada)
33
34 # Extraer los primeros 200 valores de la señal de audio
35 senial = senial[:200]
36
37 # Construir el eje del tiempo en milisegundos
38 eje_tiempo = 1000 * np.arange(0, len(senial), 1) / float(frecuencia_muestreo)
39
40 # Graficar la señal de audio
41 plt.plot(eje_tiempo, senial, color='black')
42 plt.xlabel('Tiempo (milisegundos)')
43 plt.ylabel('Amplitud')
44 plt.title('Señal de audio generada')
45 plt.show()
46
```

