

```

In [1]: 1 # SINTETIZAR TONOS
2
3 # Generar Librerías
4 import json
5 import numpy as np
6 import matplotlib.pyplot as plt
7 from scipy.io.wavfile import write
8
9 # Sintetizar el tono basado en los parámetros de entrada
10 def sintetizador_tono(frecuencia, duracion, amplitud=1.0, frecuencia_muestreo
11 # Construir el eje de tiempo
12     eje_tiempo = np.linspace(0, duracion, duracion * frecuencia_muestreo)
13
14 # Construir la señal de audio
15     senial = amplitud * np.sin(2 * np.pi * frecuencia * eje_tiempo)
16
17     return senial.astype(np.int16)
18
19 if __name__ == '__main__':
20     # Nombres de los archivos de salida
21     archivo_tono_generado = 'tono_generado.wav'
22     archivo_secuencia_tono_generada = 'secuencia_de_tono_generada.wav'
23
24     # Source: http://www.phy.mtu.edu/~suits/notefrecuencias.html
25     archivo_mapeo = 'tone_mapping.json'
26
27     # Cargue el mapa de tono a frecuencia desde el archivo de mapeo
28     with open(archivo_mapeo, 'r') as f:
29         mapa_tonos = json.loads(f.read())
30
31     # Configure los parámetros de entrada para generar el tono 'F'
32     nombre_tono = 'F'
33     duracion = 3 # segundos
34     amplitud = 12000
35     frecuencia_muestreo = 44100 # Hz
36
37     # Extrae la frecuencia del tono
38     frecuencia_tono = mapa_tonos[nombre_tono]
39     # Genere el tono usando los parámetros anteriores
40     tono_sintetizado = sintetizador_tono(frecuencia_tono, duracion, amplitud
41
42     # Escribe la señal de audio en el archivo de salida.
43     write(archivo_tono_generado, frecuencia_muestreo, tono_sintetizado)
44     # Defina la secuencia de tonos junto con las duraciones correspondientes
45     tono_secuencia = [('G', 0.4), ('D', 0.5), ('F', 0.3), ('C', 0.6), ('A',
46
47     # Construya la señal de audio basándose en la secuencia anterior
48     senial = np.array([])
49     for item in tono_secuencia:
50         # Obtiene el nombre del tono
51         nombre_tono = item[0]
52
53         # Extrae la frecuencia correspondiente del tono.
54         frecuencia = int(mapa_tonos[nombre_tono])
55         # Extrae la duración
56         duracion = item[1]

```

```
57     duracion = int(duracion)
58     # Sintetizar el tono
59     tono_sintetizado = sintetizador_tono(frecuencia, duracion, amplitud,
60     # Añadir la señal de salida
61     senial = np.append(senial, tono_sintetizado, axis=0)
62
63     # Guarda el audio en el archivo de salida
64     write(archivo_secuencia_tono_generada, frecuencia_muestreo, senial)
```