

```
In [2]: 1 # TRANSFORMACIÓN AL DOMINIO DE LA FRECUENCIA
2
3 # Importar Librerías
4 import numpy as np
5 import matplotlib.pyplot as plt
6 from scipy.io import wavfile
7
8 # Leer el archivo de audio
9 frecuencia_de_muestreo, senial = wavfile.read('palabra_hablada.wav')
10
11 # Normalizar los valores
12 senial = senial / np.power(2, 15)
13
14 # Extraer la longitud de la señal de audio
15 longitud_senial = len(senial)
16
17 # Extraer la mitad de la longitud
18 mitad_longitud = np.ceil((longitud_senial + 1) / 2.0).astype(np.int)
19
20 # Aplicar la Transformada de Fourier
21 frecuencia_senial = np.fft.fft(senial)
22
23 # Normalización
24 frecuencia_senial = abs(frecuencia_senial[0:mitad_longitud]) / longitud_senial
25
26 # Cuadrado
27 frecuencia_senial **= 2
28
29 # Extrae la longitud de la señal de frecuencia transformada
30 len_fts = len(frecuencia_senial)
31
32 # Ajustar la señal para casos pares e impares
33 if longitud_senial % 2:
34     frecuencia_senial[1:len_fts] *= 2
35 else:
36     frecuencia_senial[1:len_fts-1] *= 2
37
38 # Extraer el valor de potencia en dB
39 potencia_senial = 10 * np.log10(frecuencia_senial)
40 # Construir el eje X
41 eje_x = np.arange(0, mitad_longitud, 1) * (frecuencia_de_muestreo / longitud_senial)
42 # Graficar la figura
43 plt.figure()
44 plt.plot(eje_x, potencia_senial, color='black')
45 plt.xlabel('Frecuencia (kHz)')
46 plt.ylabel('Potencia de la señal (dB)')
47 plt.show()
```

