**Fecha Máxima de entrega: Lunes 21 de Enero.**

**Instrucciones:** Resuelva el problema propuesto usando Python. Envíe todos los archivos necesarios para reproducir sus resultados (archivos de datos, códigos .py, notebooks .ipynb, etc.) por email a `nibarra@ubiobio.cl`.

Sea $h(t)$ una función con $t \in ]-\infty, +\infty[$, se define la transformada de Fourier (FT por las siglas en inglés de *Fourier Transform*) continua $\mathcal{F}[h(t)](f) := \tilde{h}(f)$, como aquella transformación que aplicada sobre la función original, la descompone en una superposición continua de funciones periódicas con frecuencias $f$. Convencionalmente consideraremos la Transformada de Fourier $\mathcal{F}$ y su inversa $\mathcal{F}^{-1}$ como:

$$\tilde{h}(f) := \int_{-\infty}^{\infty} h(t) e^{-i2\pi f t} dt, \qquad h(t) := \int_{-\infty}^{\infty} \tilde{h}(f) e^{i2\pi f t} df. \tag{1}$$

Cuando la señal a estudiar corresponde a datos provenientes de un laboratorio o se han conseguido simulando algún fenómeno físico en el computador, no se tiene una expresión analítica a evaluar en la ec. (1)(a), sino que se tiene un conjunto de $N$ pares de números $(t_j, h_j)$, con $j = 0, 1, 2, \ldots, N-1$. Pese a ello, considerando la extensión periódica de la señal muestreada durante un tiempo $T$ (tiempo total del muestreo), igual podemos definir la Transformada de Fourier Discreta (DFT por las siglas en inglés de *Discrete Fourier Transform*) como se indicará a continuación.

Si la señal está uniformemente distribuida en el tiempo, tal que $t_j = j\Delta t$ con $j = 0, 1, \ldots$ y $\Delta t$ el espaciamiento temporal, entonces se define la Transformada de Fourier discreta de la señal y su inversa como

$$\tilde{h}_k := \frac{1}{N} \sum_{j=0}^{N-1} h_j e^{-i2\pi kj/N}, \qquad h_j := \sum_{k=0}^{N-1} \tilde{h}_k e^{i2\pi kj/N}. \tag{2}$$

Note que la ec. (2)(a) se aproxima a la integral de Riemann de (1)(a), con $T = N\Delta t$ y $f_k := k/T$.

1. Abra un Jupyter Notebook y copie el siguiente código:

```
import numpy as np #Se carga NumPy
import matplotlib.pyplot as plt #Se carga matplotlib.pyplot
import IPython.display as ipd #Se carga Ipython.display
from scipy.fftpack import fft, fftfreq #Carga Discrete Fourier Transform
from scipy.io import wavfile #Se carga modulo para leer y escribir archivos en formato wav

def fft_local(t,y):
    fourier = fft(y)
    freq = fftfreq(len(t), np.diff(t)[0])
    mask = np.where(freq>=0)[0]
    return [freq[mask], fourier[mask]]

tmax = 5. # Tiempo max. en segundos
fs = 22050 # Tasa de sampleo
t = np.linspace(-tmax,tmax,int(tmax*fs)) #arreglo de tiempo
dt = np.diff(t)[0]
```

2. Con NumPy genere una señal periódica en la forma

$$y_1(t) = A_1 \cos(2\pi f_1 t), \tag{3}$$

eligiendo arbitrariamente el valor de $A_1$ y $f_1$ (ambos positivos, se sugiere $A_1 = 1$ y $f_1 = 400$ Hz). Con la función `plt.plot` grafique $y_1(t)$ versus $t$ y con la función `ipd.Audio` resproduzca el sonido de la señal generada pasando el párametro `rate=fs` (tasa de sampleo).

3. Evalúe la DFT mediante la función `fff_local` y grafique la magnitud de la transformada fourier versus la frecuencia.

4. Repita los puntos 2 y 3 para una señal

$$y_2(t) = A_2 \cos(2\pi f_2 t). \tag{4}$$

Se sugiere elegir $A_2 = A_1$ y $f_2 = 2\,000$ Hz.

5. Repita los puntos 2 y 3 para la señal

$$y_3(t) = y_1(t) + y_2(t). \tag{5}$$

6. Desde aquí descargue el archivo de audio `CongaGroove-mono.wav` y cárguelo con la función `wavfile.read` copiando el siguiente código:

```
ffss, y4 = wavfile.read('./CongaGroove-mono.wav')
```

Genere el arreglo de temporal asociado copiando el siguiente código:

```
dt4 = 1/fs4
N4 = len(y4) #numero de puntos
fN = 1./(2.*dt4) #frecuencia de Niquist
t4 = np.linspace(0, dt4*N4, N4) #arreglo temporal
```

Con la función `plt.plot` grafique $y_4(t_4)$ versus $t_4$ y con la función `ipd.Audio` resproduzca el sonido de la señal cargada pasando el párametro `rate=fs4` (tasa de sampleo). Repita el punto 3 para la señal $y_4(t_4)$.

7. Observe sus gráficos, analízelos y en su correo responda: ¿Qué información de una señal le entrega su respectiva DFT?