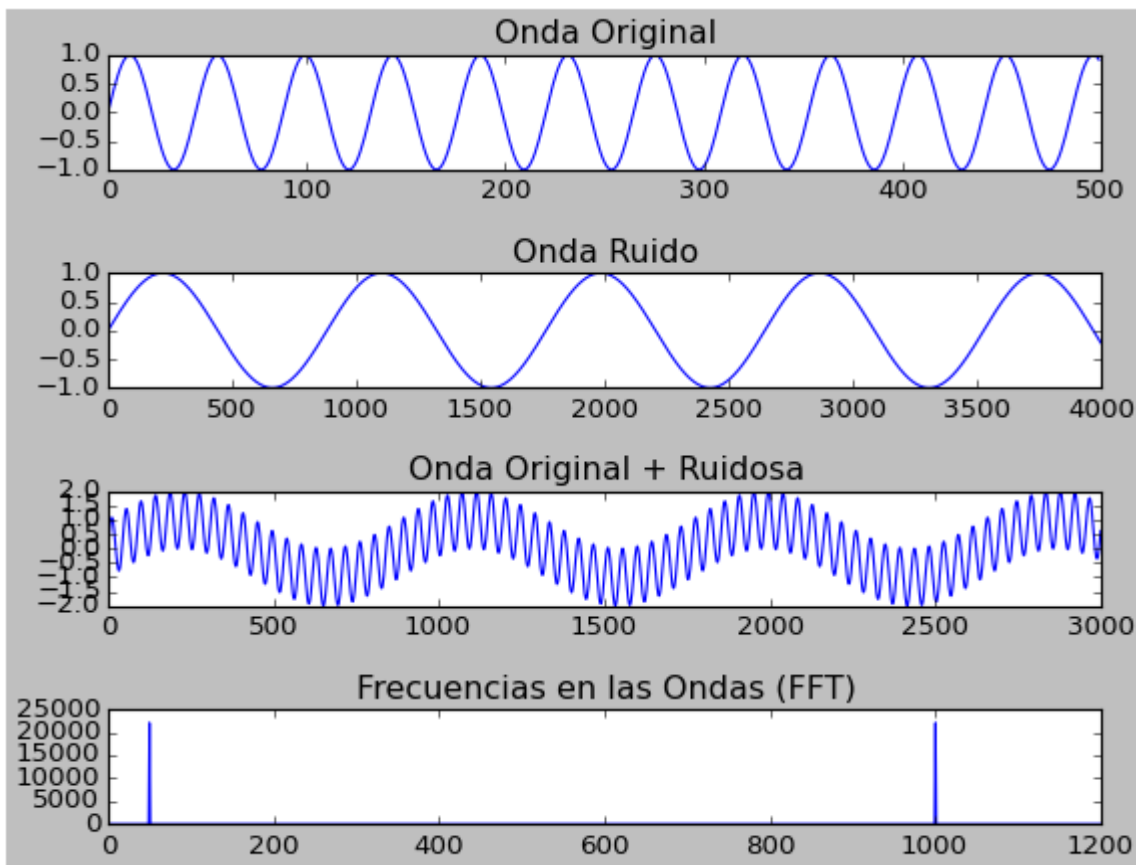


## Lab #2 Evaluado – INFO1155

By Alberto Caro

1.- El manejo de señales es fundamental en **Arquitectura de Hardware**. Este problema tiene que ver con la aplicación de la transformada rápida de **Fourier (FFT)**. La **FFT** se utiliza principalmente para obtener las frecuencias presentes en una señal. Utilizando **numpy** se puede calcular la **FFT**. Utilizando el siguiente script obtenga las gráficas y la **FFT** de las señales. Explique claramente sus resultados.

```
1 import numpy as np, matplotlib.pyplot as plt
2
3 FREQ_0 = 1000 # Frecuencia main
4 FREQ_1 = 50 # Frecuencia Ruido
5 SAMPLE = 44100 # Muestras por Segundo
6 S_RATE = 44100.0 # Tasa de Muestreo
7
8 s_1 = [np.sin(2*np.pi * FREQ_0 * i/S_RATE) for i in range(SAMPLE)] # 1000 Senos c/1s
9 s_2 = [np.sin(2*np.pi * FREQ_1 * i/S_RATE) for i in range(SAMPLE)] # 50 Senos c/1s
10 w_1 = np.array(s_1) ; w_2 = np.array(s_2) # Listas 2 Array
11 w12 = w_1 + w_2 # Sumamos 2 ondas
12
```



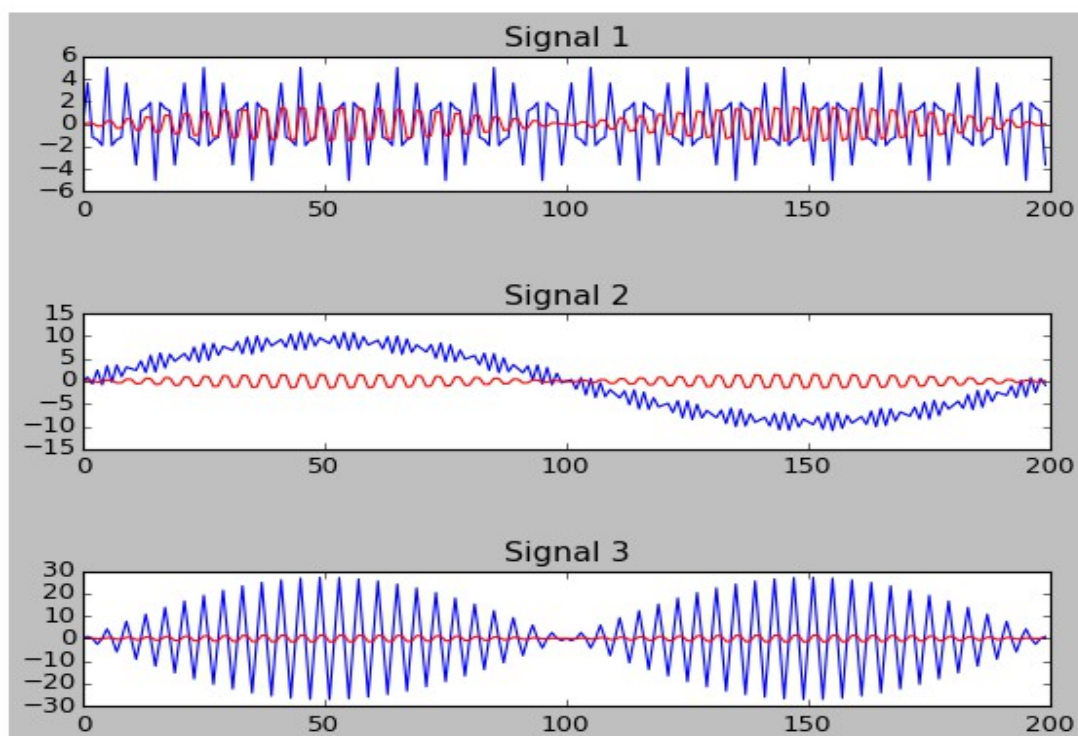
¡Investigar FFT con numpy!

2.- Utilizando el siguiente script grafique los datos mediante **Matplotlib** de **Python** utilizando el **Filtro Complementario** aplicados a las señales **aS[0]**, **aS[1]** y **aS[2]**. Explique claramente sus resultados.

```

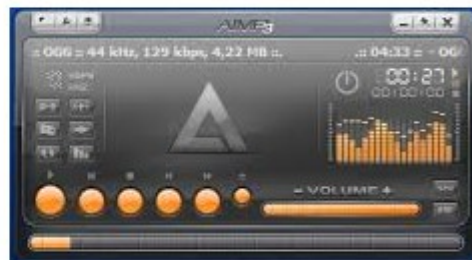
2 import numpy as np, matplotlib.pyplot as plt
3 FREQ_0 = 9000 ; FREQ_1 = 5000; FREQ_2 = 100 # Frecuencia 1, 2 y 3
4 SAMPLE = 20000 ; S_RATE = 20000.0 # Samples y Tasa de Muestreo
5 nMAX = 20000
6 # Ondas....
7 aW = [
8     [2*np.sin(2*np.pi * FREQ_0 * i/S_RATE) for i in range(SAMPLE)],
9     [3*np.sin(2*np.pi * FREQ_1 * i/S_RATE) for i in range(SAMPLE)],
10    [9*np.sin(2*np.pi * FREQ_2 * i/S_RATE) for i in range(SAMPLE)]
11 ]
12 #Signals...
13 aS = [
14     np.array(aW[0]) + np.array(aW[1]),
15     np.array(aW[0]) + np.array(aW[2]),
16     np.array(aW[1]) * np.array(aW[2])
17 ]
18 def Filter_Comp(aV,nA):
19     aF[0] = aV[0]
20     for i in range(1,nMAX):
21         aF[i] = nA * aV[i] + (1.0 - nA) * aF[i-1]
22     return aF
23

```



En las gráficas anteriores, las curvas **azules** son señales de **aS** originales y las **rojas** son las señales **aS** con **Filtro Complementario**.

3.- Realice un programa en **Python** que controle el funcionamiento del **WIMAMP** o **AIMP** mediante el control de sus teclas asignadas para: **PLAY, STOP, VOLUMEN +, VOLUMEN-, NEXT, PREVIOUS, PAUSE**, etc. Utilice **Win32Api** de **Python** con el método **User32.Keybd\_event()** para simular la presión y liberación de las teclas. Implemente un protocolo de comunicación serial mediante la utilización de **VSPE**. La aplicación cliente envía los comandos a la aplicación server que controla el reproductor de audio. Sea creativo e investigue afondo . Debe tener a lo menos **10 archivos mp3/mp4** para ser reproducidos por el media player de su elección.



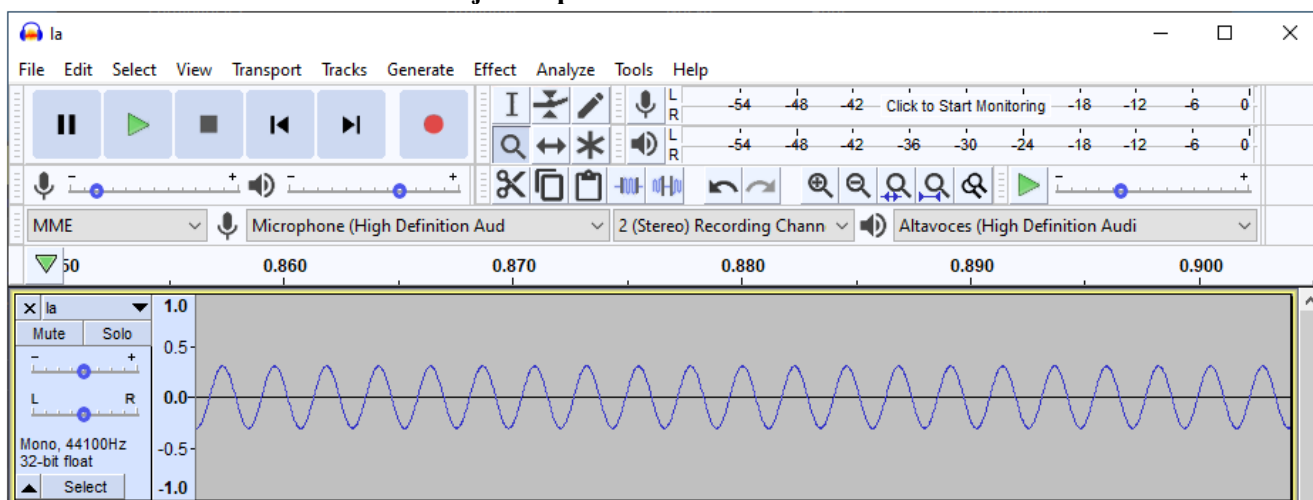
4.- Utilizando el módulo **Wave** de **Python**, genere los siguientes archivos de sonido **[.WAV]** . Cada nota debe durar **1** segundo. Investigue y utilice **pack** y **unpack** del módulo **struct** para preparar los datos de audio. Explique claramente sus resultados.

1. Escala **Musical Pentatónica: Do, Re, Mi, Fa, Sol, La y Si**, a una tasa de sampleo de **44.100** en **Mono**.
2. Escala **Musical Pentatónica: Si, La, Sol, Fa, Mi, Re y Do**, a una tasa de sampleo de **22.050** en **Stereo**.
3. Escala **Musical Pentatónica: Do, Re, Mi, Fa, Sol, La y Si**, a una tasa de sampleo de **8.000** en **Mono**.
4. Genere la siguiente onda en stereo (**RATE=44.100**) durante **10s** y visualice con **Audacity**:  

$$y = 8.000 * \sin(2 * \pi * 500.0 / \text{RATE} * i) + 8.000 * \sin(2 * \pi * 250.0 / \text{RATE} * i), i = 0, 1, \dots, \text{RATE}$$
5. Baje el volumen de la onda anterior en un **75%** utilizando **Python**.
6. Limpie el canal izquierdo de la señal anterior con **Python** y reproduzca con **Audacity**.
7. Utilice el software **Audacity** para analizar y reproducir los sonidos.

## ¡Investigar!

Trabajo Grupo de 2 Personas o Individual



Fecha de entrega y defensa en oficina profesor: **Miércoles 19 Junio** desde las **14:00 – 18:00**. Informe Impreso