

Actividad: Detección de la interfaz única y del promedio de conjunto

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import psycopg2 as ps
```

```
/usr/local/lib/python3.6/dist-packages/psycopg2/__init__.py:144: UserWarning: The psycopg2 module is deprecated. Please use the psycopg2 module instead.
  """
```

```
mod3_xlsx = "https://github.com/dhinojosac/mit_beyond_iot/blob/master/mod3/tof_act1.csv?raw=t"
df = pd.read_csv(mod3_xlsx, header=None)
df = df.transpose()
df.shape

(100, 25000)
```

2) Utilizando los datos obtenidos, determine la frecuencia de

- muestreo del DAQ (sabiendo que el tiempo total de escucha es de 840 microsegundos).

Comprobación 1: ¿Cuál es la frecuencia de muestreo del DAQ? ¿Es lo suficientemente alta para capturar datos?

```
Te = 840e-06 # Tiempo total de escucha 840 [us]
print("Tiempo de escucha: %f [s]"%Te)

Ts = Te/(df.shape[1]-1)
print("Tiempo de muestreo Ts: %.10f [s]"% Ts)
fs = 1/Ts
print("Frecuencia de muestreo fs: %.2f [Hz]"% fs)

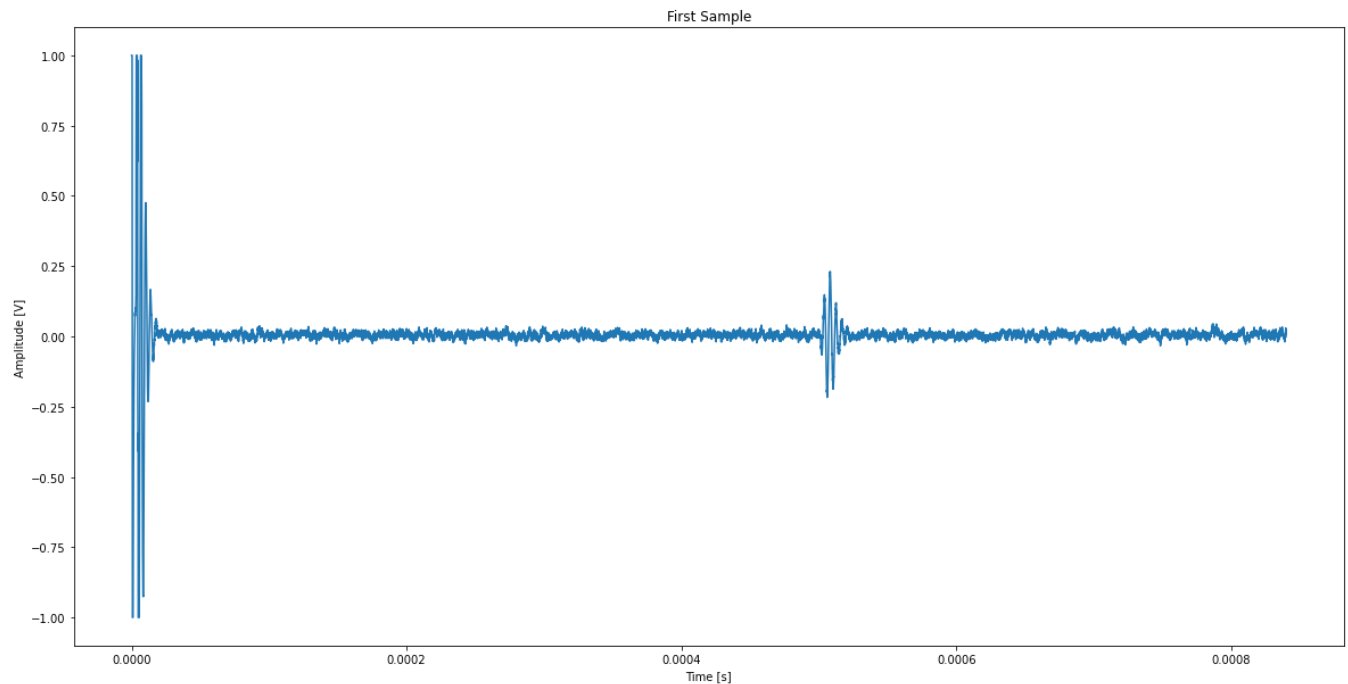
Tiempo de escucha: 0.000840 [s]
Tiempo de muestreo Ts: 0.0000000336 [s]
Frecuencia de muestreo fs: 29760714.29 [Hz]
```

```
# muestra de la primera lectura
```

```
N = df.shape[1] # Numero de muestras
```

```
x = np.linspace(0.0, N*Ts, N)

plt.figure(figsize=(20,10))
plt.plot(x,df.iloc[0])
plt.title("First Sample")
plt.xlabel("Time [s]")
plt.ylabel("Amplitude [V]")
plt.show()
```



La frecuencia de muestreo es de **29,76 MHz**. Es una frecuencia muy alta.

3) Elija un solo escaneo entre los 100 que tiene (100 columnas de la matriz de datos), trace la amplitud (que es el voltaje registrado por el

- transductor) en función de la distancia, suponga que la velocidad del sonido en el aire es de aproximadamente 343 m/s.

```
N = df.shape[1] # Numero de muestras

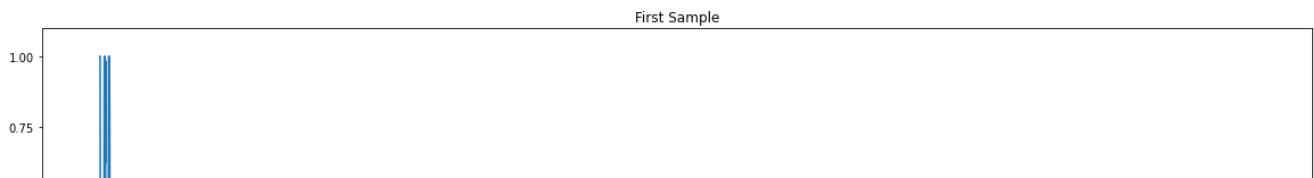
x = np.linspace(0.0, N*Ts, N)

vs = 343 # velocidad del sonido en m/s

d = np.linspace(0.0, N*Ts*vs*0.5, N)

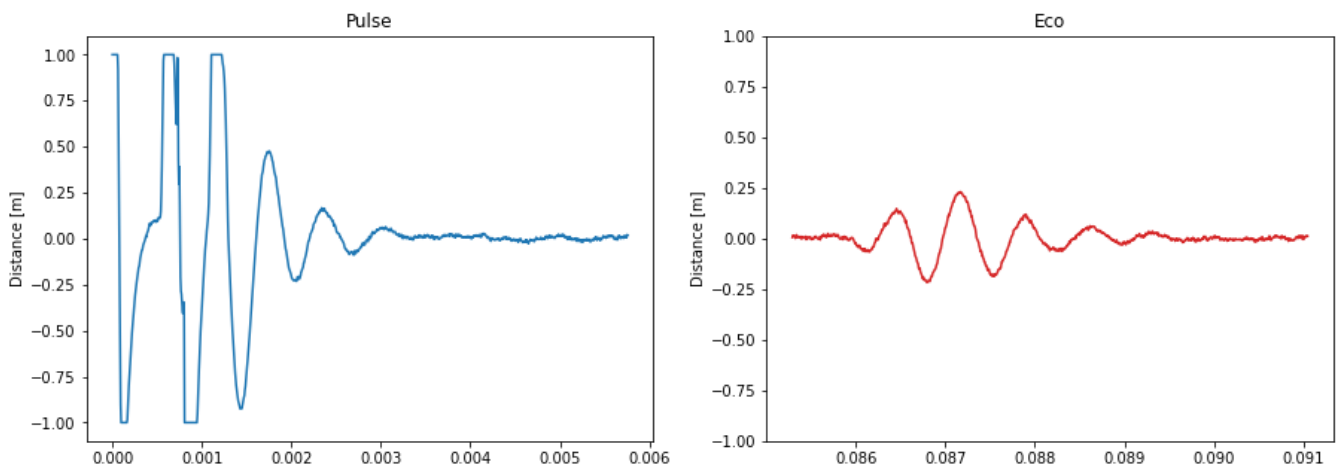
sample = df.iloc[0]

plt.figure(figsize=(20,10))
plt.plot(d,sample)
plt.title("First Sample")
plt.xlabel("Distance [m]")
plt.ylabel("Amplitude [V]")
plt.show()
```



```
# PULSO
fig, ax = plt.subplots(1,2, figsize=(15,5))
ax[0].plot(d[0:1000],sample[0:1000])
ax[0].set_title('Pulse')
ax[0].set_ylabel('Distance [m]')

ax[1].plot(d[14800:15800],sample[14800:15800], 'tab:red')
ax[1].set_title('Eco')
ax[1].set_ylabel('Distance [m]')
ax[1].set_ylim([-1,1])
plt.show()
```



Comprobación 2

- ¿Cuál es la distancia hasta la superficie de la interfaz?

Como se aprecia en los gráficos anteriores, se grafico amplitud vs distancia. Para esto se tuvo en cuenta la velocidad del sonido que para este ejemplo era 343 [m/s]. Con esto se multiplica el tiempo de cada muestra por la velocidad y se tiene la distancia. Pero para obtener la distancia a la superficie, se dividió el tiempo a la mitad así tener un gráfico aproximado a la distancia real. Por lo que la **distancia a la superficie** es de aproximadamente **0.086 [m]**.

- ¿Cuáles son las principales fuentes de error?

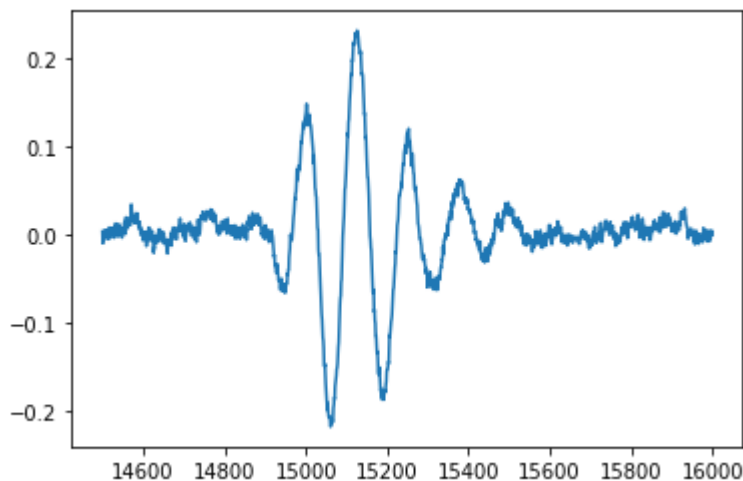
Los cambios de material por las que el sonido se propaga tienen diferentes velocidad, por lo que asumir que es 343 m/s en todos los materiales agrega error. La velocidad del sonido también se ve

afectada por la temperatura de los materiales por las que se propaga por lo que también agrega error.

4) Usando el conjunto de datos de (3), trunque los datos para que solo incluyan el eco y no el pulso inicial.

```
eco = df.iloc[0,14500:16000]
eco.plot()
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f94f35b42b0>



5) Para el mismo conjunto de datos en (4), obtenga el espectro de frecuencia asociado. Por favor, siguiendo las instrucciones del apéndice:

```
from scipy.fftpack import fft,fftfreq

# Numero de muestras
N = len(eco)

# Espacio de muestreo
T = Ts
x = np.linspace(0.0, N*T, N)
y = eco.to_numpy()
yf = fft(y)

xf = fftfreq(N,Ts)

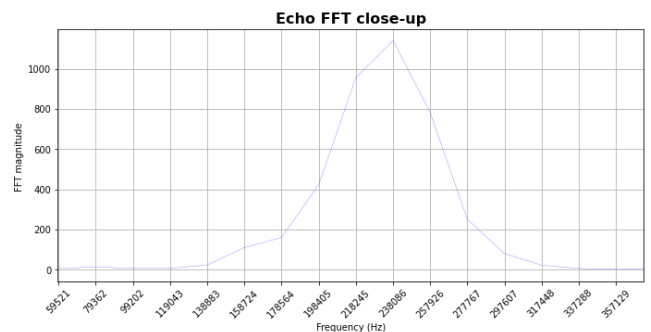
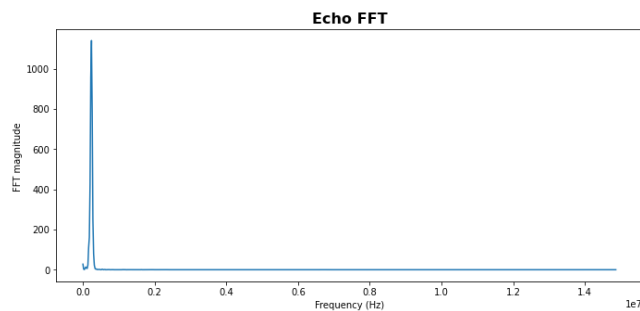
fig = plt.figure(figsize=[25,5])
```

```

ax = plt.subplot(121)
ax.plot(xf[:N//2], np.abs(yf[:N//2])*np.abs(yf[:N//2]))
ax.set_title('Echo FFT', fontsize= 16, fontweight="bold")
ax.set_ylabel('FFT magnitude ')
ax.set_xlabel('Frequency (Hz)')

ax2 = fig.add_subplot(122)
ax2.plot(xf[:N//2], np.abs(yf[:N//2])*np.abs(yf[:N//2])), lw=0.2, c='b')
ax2.set_xticks(xf)
plt.xticks(rotation=45)
ax2.set_xlim(59000,int(1/(80*Ts)))
ax2.set_title('Echo FFT close-up', fontsize= 16, fontweight="bold")
ax2.set_ylabel('FFT magnitude ')
ax2.set_xlabel('Frequency (Hz)')
ax2.grid()

```



Comprobación 3:

- ¿Cuál es la frecuencia central del transductor?

Lo que se alcanza a ver de la FFT es que la frecuencia central del transductor es de **238 kHz**.

- ¿Cuál es su ancho de banda -6dB?

La caída de 3dB es la mitad, por lo que una caída de 6dB es cuando cae aproximadamente 1/4 la amplitud, por lo que el ancho de aproximadamente **200 kHz**.

6) Con los 100 escaneos originales, haga lo siguiente:

1. Construya un conjunto de datos promedio a partir de los 100 escaneos que tiene.
2. Con el MISMO conjunto de datos de (a), trunque los datos como hizo en (4)
3. Construya el espectro de frecuencia del conjunto de datos anterior como lo hizo en (5).
4. Superponga el espectro de frecuencia que acaba de obtener con el obtenido en (5).

Comprobación 4: ¿Qué efecto tiene el promedio en el espectro de frecuencia de una señal?

▼ Extra

```
n1 = df.iloc[:,0:2000]
n2 = df.iloc[:,14600:15600]
#ndata = pd.concat([n1,n2],axis=1)
ndata = df

f = plt.figure(figsize=(20, 20))
plt.matshow(ndata, fignum=100, cmap="gray")
plt.gca().set_aspect('auto')
cb = plt.colorbar()

#from google.colab import files
plt.savefig('reflection.pdf',dpi=1000)
files.download('reflection.pdf')
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-9-9a1c3b723a5b> in <module>()
      12 #from google.colab import files
      13 plt.savefig('reflection.pdf',dpi=1000)
----> 14 files.download('reflection.pdf')
```

NameError: name 'files' is not defined

SEARCH STACK OVERFLOW

<Figure size 1440x1440 with 0 Axes>

