Funciones

SWEEP: Genera un sine-sweep logarítmico y ademas permite guardarlo (en formato .wav). Debemos entregarle: la duracion en tiempo, las frecuencias donde comienza y termina el sweep, la frecuencia de sampleo y si queremos guardarlo o no y donde lo haremos.

Nos devuelve una tupla compuesta de los valores en tiempo y los valores en amplitud del sweep.

INVERSE\_FILTER: Toma el sweep y genera un filtro inverso a partir de este. Nos devuelve una tupla compuesta de los valores en tiempo y los valores en amplitud del filtro inverso.

LIST\_UDIM: Transforma una matriz en una lista horizontal.

FREQUENCY\_LABELS: Convierte una lista de valores de tipo entero en otra lista de valores pero ahora de tipo string y con un sufijo.

GRAPHER: Permite graficar distintas señales que se ingresen como lista de tuplas, en cada tupla debe ir el valor del eje x y el eje y.

Obligatoriamente se debe seleccionar la dimensión del lienzo, si es uno solo (1,1) y si son varios (2,2), (1,2) etc.

Opcionalmente se puede definir si se quiere imprimir un gráfico: normal, logarítmico (semilogx = True) o de barras (bars = True), si es logarítmico o de barras se puede seleccionar que tipo de eje x tendrá, por octavas o por tercios de octavas.

Otros parámetros opcionales se pueden pasar son el figsize (tamaño del lienzo), fontsize (tamaño de fuente que se escala según si se trata de un título, eje o leyendas), los labels de los ejes, y los límites de los ejes.

FFT: Transforma una señal respecto al tiempo a una en el dominio de las frecuencias, utilizando la Transformada Discreta de Fourier. Ingresamos la señal respecto al tiempo y nos devuelve la magnitud de la señal en el dominio de las frecuencias.

IMPULSE\_RESPONSE: Determina la respuesta al impulso de un sistema utilizando la transformada y la anti transformada discreta de Fourier. El array de magnitudes que debemos ingresar a la función deben tener la misma dimensión. Además debemos indicarle también la frecuencia de muestreo que debe utilizar. Nos devuelve una tupla que contiene la respuesta al impulso del sistema en tiempo y amplitud.

FREC\_SUM: Suma señales sinusoidales. Debemos ingresar las frecuencias que tendrán los senos a sumar, el número de muestras que tendrá la señal resultante y la duración de la misma. Nos devuelve una tupla que contiene los valores en tiempo y la magnitud de la señal resultante.

MAKE\_LIST: Convierte toda variable entregada en una lista.

ROUND\_ARRAY: Redondea el float ingresado o los floats del array ingresado tomando como número de cifras significativas un valor ingresado.

F\_M: Calcula la frecuencia central de un determinado array, siguiendo con la normativa UNE-EN 61260.

F\_C: Determian la frecuencia de corte de una frecuencia central dada, siguiendo con la normativa UNE-EN 61260.

BANDPASS\_FILTER: Genera un filtro pasa-banda, siguiendo con la normativa UNE-EN 61260. Debemos entregarle, las bandas de frecuencias centrales, la frec de muestreo, el ancho de banda, el octave ratio, el orden del filtro.

MEAN: Calcula la media de un vector dado.

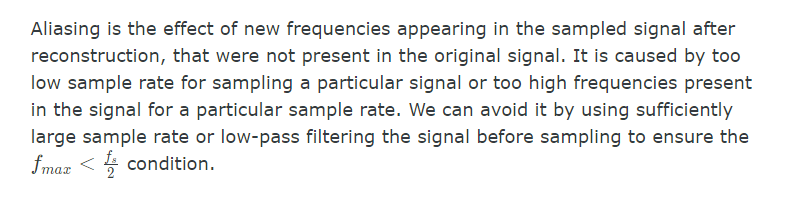
SPL: Calcula el nivel de presión sonora de un float o de un array.

SPL\_AVE: Calcula el promedio del nivel de presión sonora.

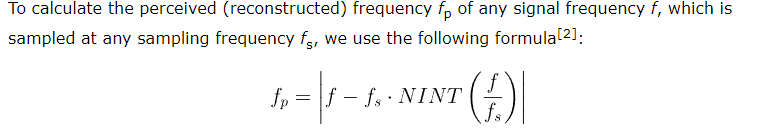
FILTER\_BANK:

Genera un banco de filtros con los coeficientes sos del método signal.butter() para cada banda. Luego aplica cada filtro a un audio dado para calcular el SPL promedio de cada banda. Devuelve el vector de coeficientes sos, las bandas, y el vector de promedios.

Punto 4:



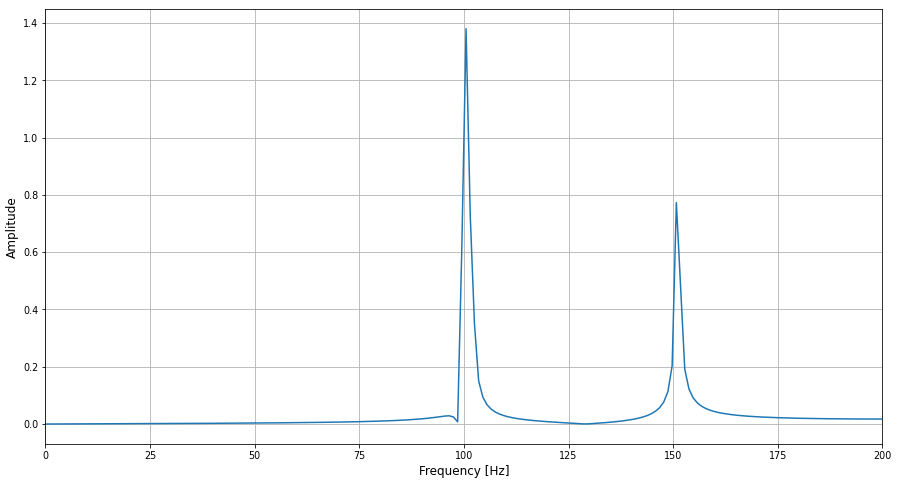
El aliasing es el efecto de aparición de nuevas frecuencias en la señal sampleada luego de su reconstrucción, que no estaban presentes en la señal original. Esto es causado por la elección de una frecuencia de muestreo demasiado baja para una señal en particular o también debido a la presencia de altas frecuencias en la señal. Podemos evitar el aliasing utilizando una frecuencia de muestreo suficientemente grande o bien pasando la señal por un filtro pasa-bajos para asegurarnos que al momento de muestrearla cumpla la condición de Nyquist.





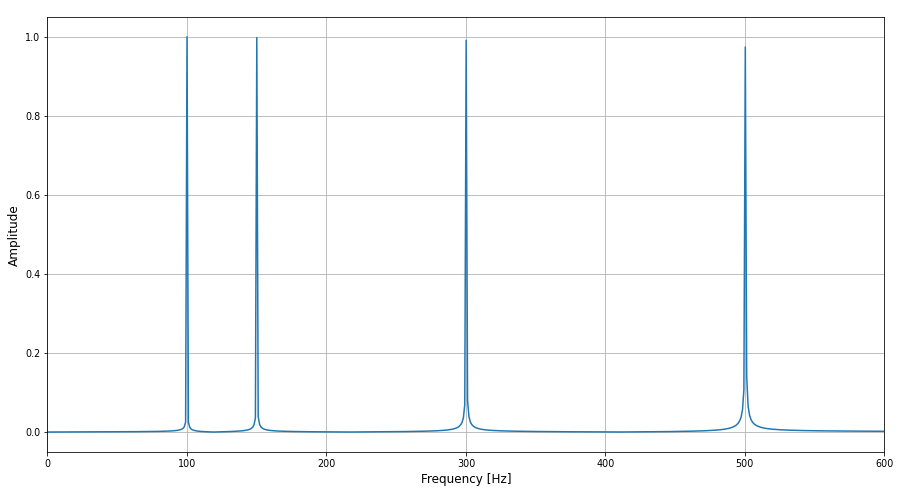
En la que NINT es el entero más cercano realizando un redondeo hacia arriba. Por ejemplo 10.5 se redondea a 11 y 10.49 se redondea a 10

En el caso de 300hz y 500hz, siendo 400hz la frecuencia de muestreo, da 100hz y 100hz respectivamente, por lo tanto vemos esto:



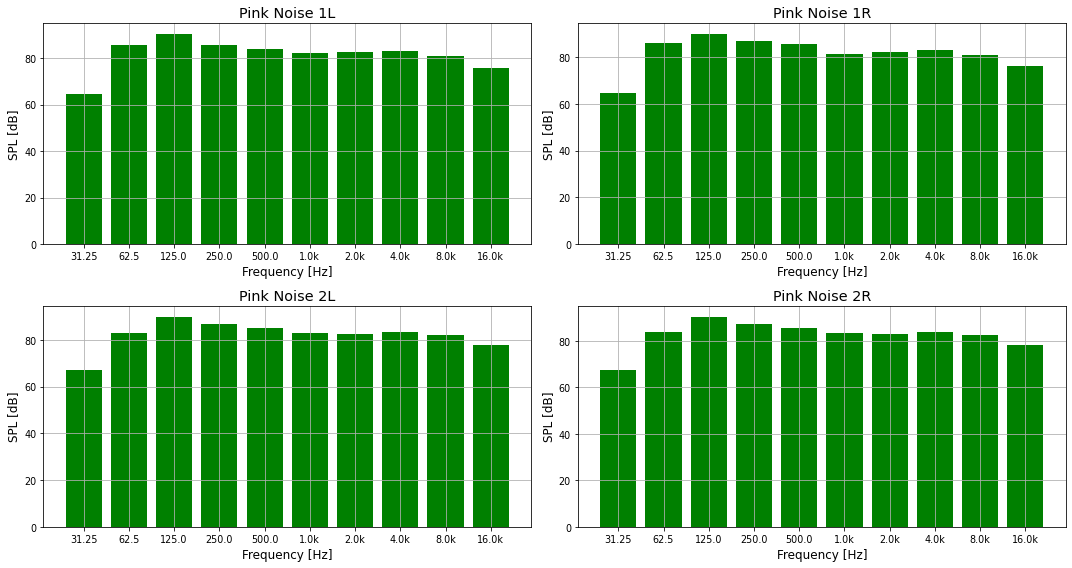
Se ve que la amplitud es mayor que 1dB, que es lo que contiene la señal original, además de que desaparecieron las frecuencias 300hz y 500hz.

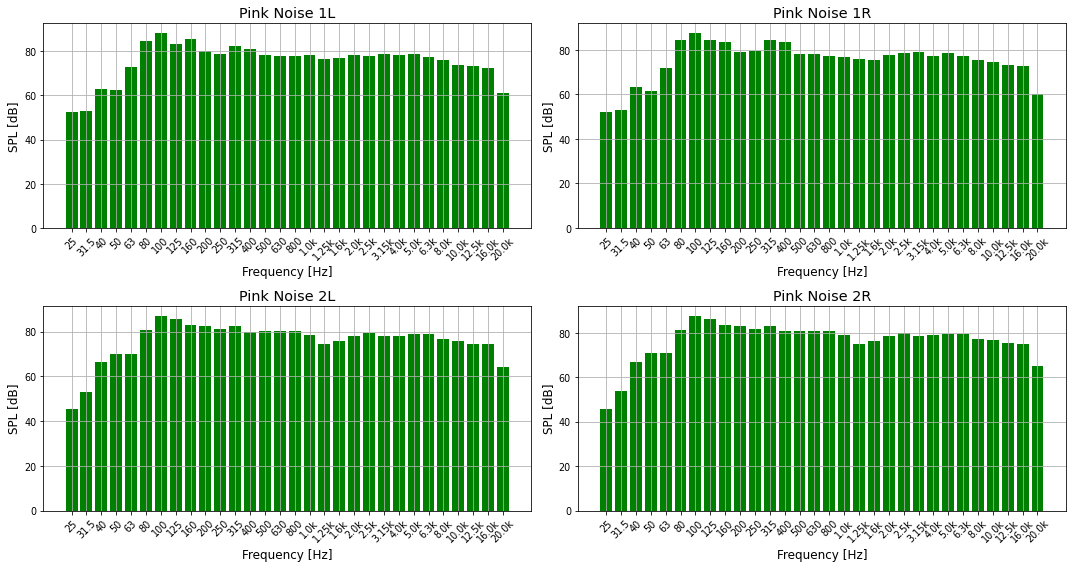
La señal original es esta:

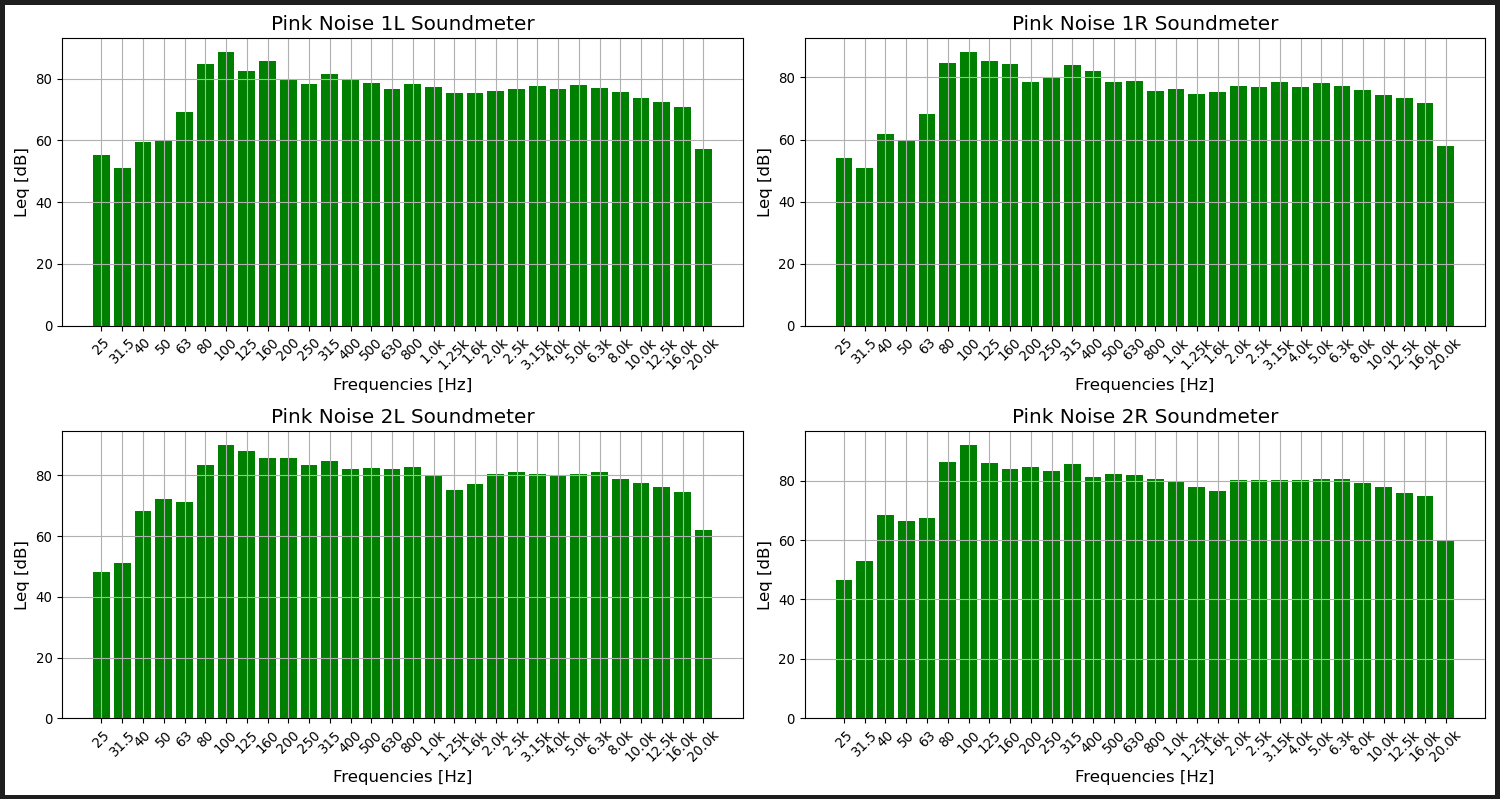


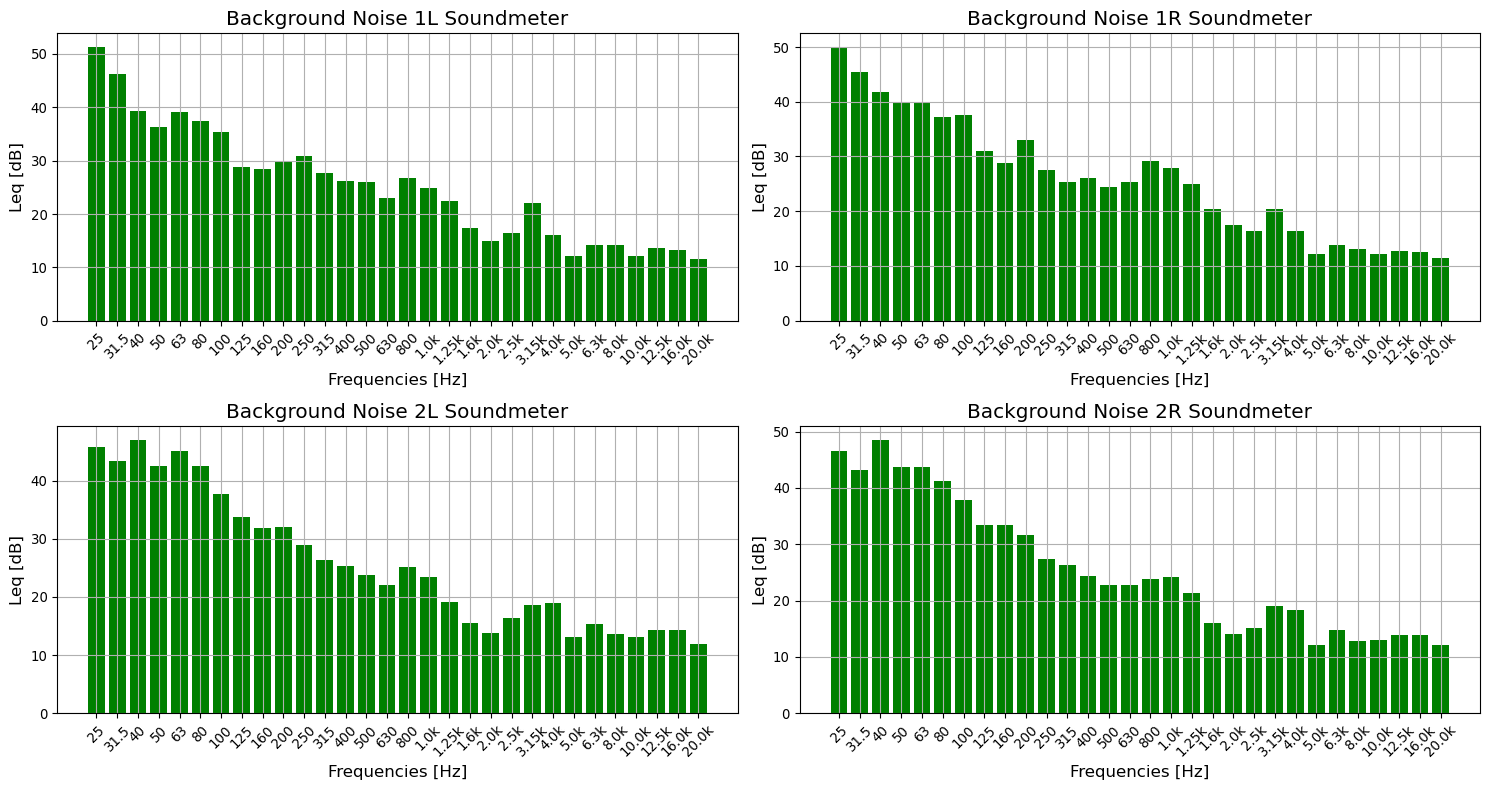
Obtenida con 4000 muestras por segundo, una frecuencia de muestreo que supera ampliamente lo requerido por el teorema de Nyquist

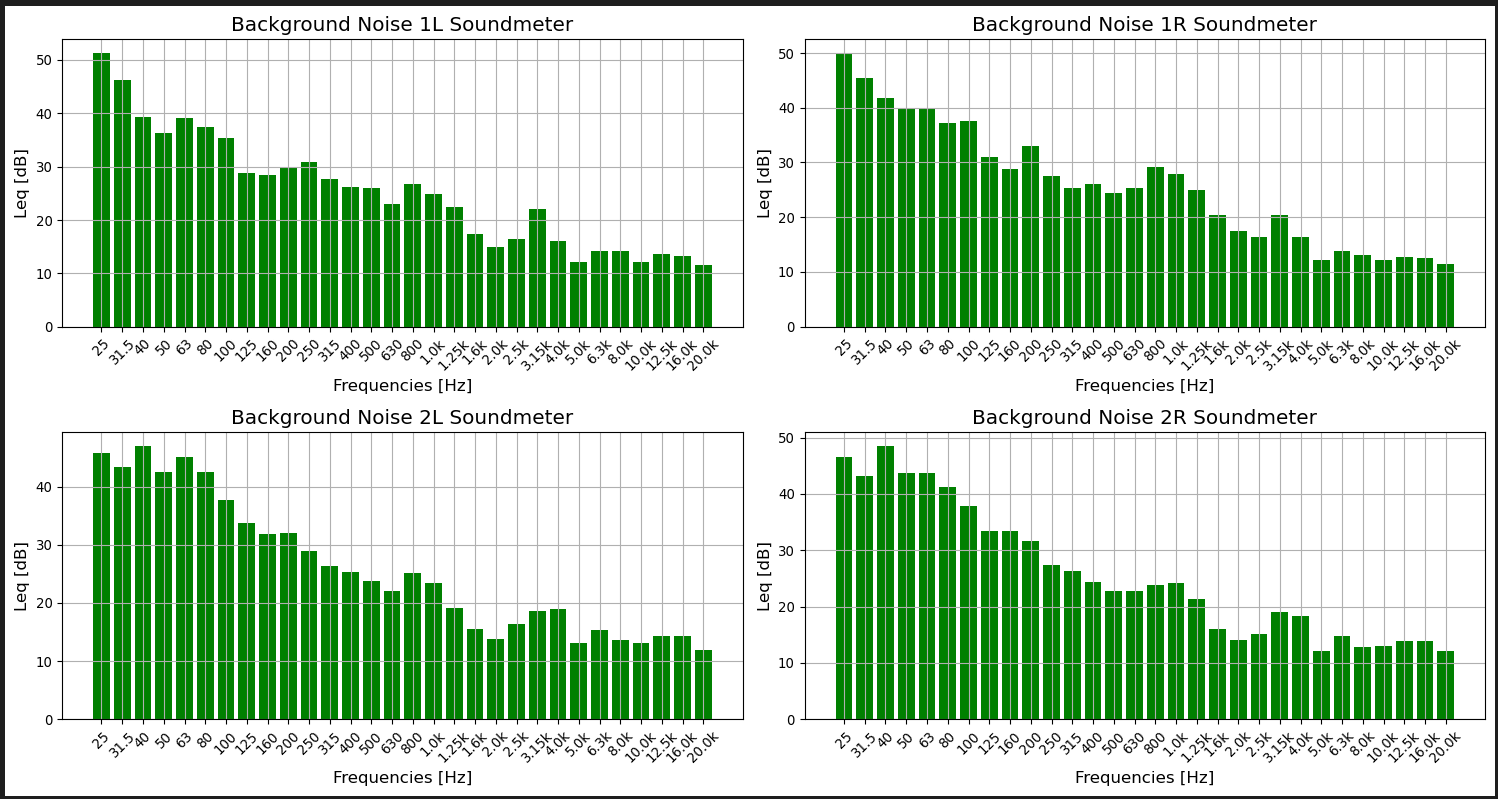
# Analisis de graficos:











## Exposición

1. (2 min) Dia de medición:
2. (5 min) Mediciones de ruido de fondo y rosa:
3. (4 min) Respuesta al impulso y en frecuencia
4. (4 min) Alias

Guión:

## Punto 1 - Dia de medición 26/10/2022, aula taller de ing en sonido, piso 2 de caseros 2 (Joaco)

Empezamos colocando la fuente en el lugar donde se ubica el profesor y los receptores (micrófonos y sonómetros) donde se ubican los alumnos, **\*mostrar dibujos autocad\*** con el objetivo de que los resultados obtenidos simulen una situación real y cotidiana.

Una vez colocados los equipamientos **\*mostrar equipamientos\***, se grabó el ruido de fondo y se reprodujeron los audios pedidos por consigna **\*ruido rosa y sine sweep\*.**

Ruido de fondo y ruido rosa se grabó con los micrófonos y se midió con el sonómetro el nivel continuo equivalente. Posterior a esto el sine sweep generado por código, se grabó sólo con los micrófonos.

El día de la medición tuvimos las siguientes variables termodinámicas **\*mostrar tabla de variables\***, que formaron parte de los resultados conseguidos los cuales veremos a continuación.

## Punto 2 - Banco de filtros (5 min) Maxi

A partir de la norma UNE 61260 se definieron funciones para calcular la frecuencia central y el ancho de banda de filtros pasabanda. Según define la norma, se consideró un cociente de octava G2, una frecuencia de referencia de 1 kHz y un orden de N=4 de manera predeterminada pero estos son parámetros modificables en todas las funciones.

Haciendo uso del módulo Signal de SciPy se definió bandpass\_filter() que es una función que devuelve los coeficientes sos así como la magnitud y fase de las distintas frecuencias para graficar la respuesta del filtro.

Luego, se definieron las funciones SPL() y SPL\_ave() para calcular: (1) el SPL para un valor dado de presión y (2) el nivel promedio para una señal de audio dada a partir de una referencia, respectivamente.

El valor de presión de referencia es 20uPa por 0.63, que resultó ser el nivel full scale promedio de la grabación del calibrador.

Usando este valor de referencia, se obtuvieron gráficos del **SPL en función del tiempo** para el ruido de fondo y ruido rosa en los 4 puntos de medición.

Finalmente, se diseñó un banco de filtros filter\_bank() que toma un audio array y su frecuencia de muestreo para aplicarle bandpass\_filter() a cada banda y obtener el promedio SPL de cada audio filtrado por bandas.

Por un lado, se aplicó el banco de filtros filter\_bank() diseñado por octavas y por tercios de octavas a las grabaciones de ruido de fondo y ruido rosa en sus 4 posiciones

*obteniendo los siguientes espectros*.

Y por otro lado, se analizaron los espectros de las mediciones hechas con sonómetro, observando que son consistentes con los espectros del banco de filtro diseñado. (1 solo graf)

*Las siguientes son las gráfcas de dichos espectros.*

vemos que hay una leve diferencia en altas frecuencias

vemos que se tiene 35 dB más de ruido rosa que ruido de fondo

## Punto 3 - Respuesta al impulso (4 min) Joaco ft. Abel

Para obtener la respuesta al impulso del recinto se utilizó un sweep logarítmico ya que este nos permite una correcta distribución de energía en cada banda de frecuencias.

Para lograrlo se trasladó a código la expresión matemática **\*mostrar x(t)\*** con sus respectivas constantes K y L. Se puede notar que ambas constantes están divididas por una relación logarítmica entre la frecuencia final e inicial **\*Ln(f2/f1)\***.

Los valores de frecuencia inicial y final elegidos fueron de 42 Hz y 18 kHz, tomados de la respuesta en frecuencia del parlante en el cual el fabricante especifica una variación de +-3dB.

me falto comentar que se eligió arbitrariamente la frecuencia de muestreo en 96kHz.

Y en cuanto a la duración del sweep no debía ser muy corta ya que no permitiría establecer la energía acústica necesaria para una correcta medición, por lo cual se tomaron 25 segundos de duración.

Para la obtención de la respuesta es necesario convolucionar la señal de sweep generada con la grabada, pero esta última debe pasar por un filtro inverso para compensar la caída de 3 dB por octava que se atribuye a la característica logarítmica del sine sweep. Para esto se trasladó a código el procedimiento matemático **\*mostrar ecuaciones\*** creándose las funciones *inverse\_filter* e *impulse\_response*

Se procesaron los 4 archivos grabados y se obtuvo la respuesta al impulso para cada una, además de la respuesta en frecuencia con la creación de la función *fft*, la cual regresa solo el espectro positivo de la fft. **\*mostrar gráficos de las rta al impulso y frecuencia\***

Para el gráfico de la respuesta al impulso se amplió el eje x, haciendo énfasis en la zona en la que se desarrolla el impulso.

De los gráficos de respuesta en frecuencia se puede diferenciar la zona de estudio modal y la zona de análisis estadístico. Además se puede ver que en bajas frecuencias, de 42 a 100 Hz el parlante no logra superar el ruido de fondo existente por lo cual no se logran diferenciar correctamente los modos en esa región.

Podemos notar que al cambiar la disposición de los receptores la distribución de los modos también cambia **\*señalar modo 125-160, está en la 1ra medición pero no en la 2da\***.

Como conclusión diremos que: En cada posición se pueden medir diferentes modos, mientras que en la 2da disposición se notó una menor presencia de estos.

Para la auralización se tuvo que recortar la respuesta en impulso en tiempo, ya que originalmente contiene una componente de distorsión al final la cual hacía que se aplicará un efecto de “delay” en la auralización.

## Punto 4 - Alias (4 min) Ema

**\*mostrar sumatoria\***

Bien, En esta ultima parte vamos a analizar un aspecto de la transformacion de una señal del dominio continuo al discreto. Nosotros realizamos una funcion que genera la sumatoria de varias senoidales y luego las muestreamos a diferentes frecuencias

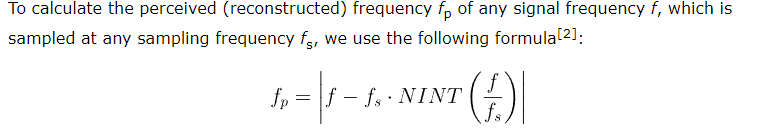
Aca podemos ver el resultado de esta sumatoria de diferentes senoidales

**\*mostrar resultado de 400hz\***

Al muestrearlas y transformarlas al dominio frecuencial con una FFT, notamos que en el de 400 muestras por segundo desaparecieron las frecuencias 300hz y 500hz, pero también tienen mayor componente de 100hz. Esto se debe al Aliasing, que ocurre cuando no se cumple el Teorema de muestreo de Nyquist-Shannon (que dice que la velocidad de muestreo debe ser el doble que la frecuencia de componente máximo de la función que está siendo muestreada).

Como consecuencia el resultado del muestreo agrega “alias” que son malas interpretaciones debido al muestreo demasiado lento. Esto puede calcularse con la siguiente formula:

**\*mostrar formula\***





En la que Fp es la frecuencia percibida final, F es la frecuencia que estamos muestreando y Fs la frecuencia de muestreo o sampling

Ademas de eso, NINT es el entero más cercano realizando un redondeo hacia arriba. Por ejemplo 10.5 se redondea a 11 y 10.49 se redondea a 10

En el caso de 300hz y 500hz, siendo 400hz la frecuencia de muestreo, nos da 100hz y 100hz en ambos casos, por lo tanto los ALIAS van a aparecer agregados en esta frecuencia de 100hz, Esto se corresponde perfectamente con el resultado obtenido

**\*mostrar resultado de 4 MIL hz\***

Siguiendo el teorema de nyquist-shannon, en el caso de 4 MIL hz de muestreo, se cumple ampliamente y el resultado es exactamente la sumatoria de las 4 senoidales que ingresamos al comienzo.

**\*mostrar gif\***

Por ultimo en este .GIF podemos ver la frecuencia en la que aparecerán los artefactos con muestreo de 400hz según la frecuencia que ingresemos: Podemos ver como al superar los 200hz comienzan a aparecer los alias y a 300hz y 500hz serán en 100hz. Esto se corresponde con los resultados de la experiencia

En la practica Si se necesita muestrear una señal que Se sabe que no se podrá cumplir el teorema de Nyquist, lo que se hace es pasarla previamente por un filtro pasa bajos para evitar la aparición de los Alias que terminarian afectando a todo el resultado

Muchas gracias por su atencion