



Desarrollo de software para el cálculo de parámetros acústicos ISO 3382

Dylan Kaplan¹, Franco Rebora², Santiago Rodríguez Salinas³

Señales & Sistemas, Ingeniería de sonido

Universidad Nacional de Tres de Febrero, Buenos Aires, Argentina.

¹ dylukaplan@gmail.com, ² reborafs@gmail.com, ³ santirodriguezsalinass@gmail.com,
1er cuatrimestre, 2020.

Resumen

El presente trabajo consta, a grandes rasgos, de realizar un programa que calcula parámetros acústicos a partir de la respuesta al impulso de un recinto. El objetivo de esto es el de realizar un aporte al campo de las ciencias acústicas utilizando nuevas tecnologías. El trabajo se desarrolla en el marco de la materia Señales y Sistemas de la Universidad de Tres de Febrero a lo largo de tres meses. Para una mejor organización del flujo de trabajo, se opta por utilizar algunos conceptos de la metodología scrum. Además, se divide el desarrollo del programa en frontend y backend para mayor practicidad. El programa consta de una aplicación web con diferentes opciones de cálculo. A la hora de obtener las señales a procesar, es requisito mínimo realizarlo con un micrófono y un parlante. A partir de analizar los resultados, se puede notar que la aplicación cumple con su función. Los resultados obtenidos concuerdan con lo esperado. Sin embargo, se encuentran diferencias con los resultados obtenidos por el módulo Aurora, desarrollado por Angelo Farina, lo cual da lugar a una próxima revisión del código y una posible segunda entrega.

Keywords: ISO 3382, IR

1. INTRODUCCIÓN

En el presente trabajo se tiene como objetivo el cálculo de parámetros acústicos de un recinto propuestos en la normativa ISO 3382, tales como EDT, T_{10} , T_{20} , T_{30} , C_{80} y D_{50} . Con este propósito, se propone diseñar el software para dicho cálculo utilizando el paradigma de programación modular con el lenguaje Python 3.

Mediante distintas funciones y métodos matemáticos el software abarca las etapas de adquisición de la señal, procesamiento y cálculo de parámetros acústicos.

Como antecedente se toma el trabajo "Desarrollo de software para el cálculo de parámetros acústicos bajo norma ISO 3382" [1], variando el software en el cual se desarrolla por el ya mencionado Python 3. Esto permite presentar un trabajo usando una herramienta más versátil y más competitiva en el mercado.

2. MARCO TEÓRICO

2.1. TIEMPO DE REVERBERACIÓN

Es el tiempo, medido en segundos, requerido para que el nivel de presión sonora decaiga 60 dB después de que la fuente sonora se haya apagado. Se puede evaluar basándose en un rango dinámico inferior a 60 dB y extrapolado a un tiempo de decrecimiento de 60 dB. Del mismo modo, si T se deriva del tiempo en el que la curva de decrecimiento alcanza primero 5 dB y 25 dB por debajo del nivel inicial, lo definimos como T_{20} . Y si se utilizan valores de decrecimiento entre 5 dB y 35 dB por debajo del nivel inicial, lo definimos como T_{30} . Este parámetro se calcula para cada banda de octava o banda de tercio de octava, en base a la norma IEC 61260 [2] y la norma ISO 3382 [3].

2.2. SINE SWEEP Y FILTRO INVERSO

El sine sweep logarítmico es un barrido sinusoidal, en el que la frecuencia instantánea varía expo-



nencialmente con el tiempo. Tiene la característica de excitar sólo una frecuencia a la vez, lo cual permite concentrar toda la energía de la señal en un ancho de banda estrecho. El filtro inverso es básicamente la señal de entrada en sí, invertido a lo largo del eje del tiempo, de modo que la frecuencia instantánea disminuye con el tiempo. En el caso logarítmico, se le agrega una modulación de amplitud para compensar diferencias de energía generada en bajas y altas frecuencias [4]. A continuación la ecuación utilizada para generar el sine sweep logarítmico:

$$x(t) = \sin[K.(e^{t/L}) - 1] \quad (1)$$

$$K = \frac{Tw_1}{\ln(\frac{w_2}{w_1})} \quad (2)$$

$$L = \frac{T}{\ln(\frac{w_2}{w_1})} \quad (3)$$

Con t la variable tiempo en segundos, T el tiempo de duración total y w_1, w_2 las frecuencias angulares inicial y final del barrido.

Con la amplitud variando respecto al tiempo, se puede ver en la Figura 1 el gráfico del sine sweep logarítmico.

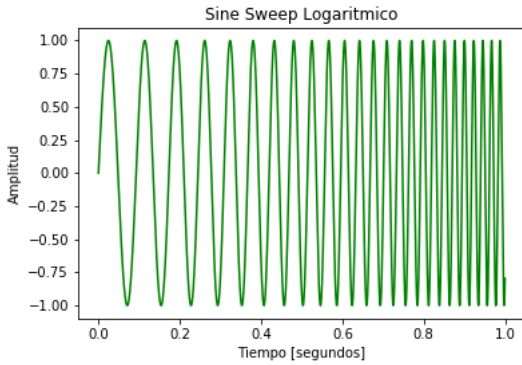


Figura 1: Gráfico de la amplitud respecto al tiempo del sine sweep logarítmico.

Y la ecuación utilizada para generar el filtro inverso:

$$k(t) = m(t)x(-t) \quad (4)$$

$$m(t) = \frac{w_1}{2\pi w(t)} \quad (5)$$

$$w(t) = \frac{d[\theta(t)]}{dt} = \frac{K}{L} e^{\frac{t}{L}} \quad (6)$$

Siendo $k(t)$ el filtro inverso, $x(t)$ el sine sweep, $m(t)$ la modulación y $w(t)$ la frecuencia instantánea.

Con la amplitud variando respecto al tiempo, se puede ver en la Figura 2 el gráfico del filtro inverso.

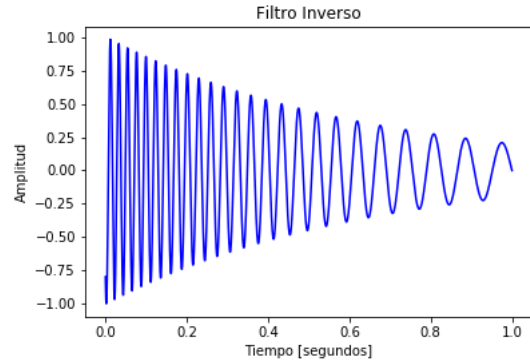


Figura 2: Gráfico de la amplitud respecto al tiempo del filtro inverso.

2.3. RUIDO ROSA

El ruido rosa es un ruido con una distribución de frecuencias tal que su densidad espectral de potencia es proporcional a la inversa de la frecuencia. Esto implica que su nivel, por bandas de tercio de octava, es constante. Por el contrario, el nivel del ruido rosa por intervalo fijo de frecuencias decae a razón de 3dB por octava, de forma no lineal aunque las frecuencias se representen en un eje lineal [5]. En la figura 3 se puede ver cómo varía la amplitud respecto a la frecuencia del ruido rosa.

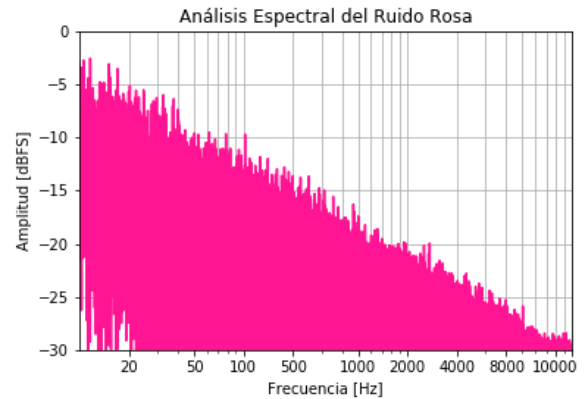


Figura 3: Análisis espectral del ruido rosa.

2.4. RESPUESTA AL IMPULSO

Se define como la respuesta de un sistema ante una señal de entrada impulsiva. Esta señal contiene toda la información que existe acerca del comportamiento del sistema, permitiendo además el cálculo de la señal de salida del sistema ante cualquier señal de entrada [6]. La sintetización del mismo se

realiza a partir de la siguiente ecuación, para una frecuencia central f_i :

$$y_i = A_i e^{(\pi_i t)} \cos(2\pi f_i t) \quad (7)$$

$$\pi_i = \frac{-\ln(10^{-3})}{T_{60}} \quad (8)$$

Luego, se obtiene la suma de varias frecuencias centrales con:

$$y = \sum_{i=1}^n y_i \quad (9)$$

2.5. ÍNDICES DE CLARIDAD Y DEFINICIÓN

C_{80} , generalmente llamado claridad, es un valor objetivo que estima la relación de energía dentro de los primeros 80 ms frente a la energía del sonido posterior. Suele usarse en actuaciones musicales. Se representa con la siguiente ecuación:

$$C_{80} = \frac{\int_{0ms}^{80ms} p^2(t) dt}{\int_{80ms}^{\infty} p^2(t) dt} \quad (10)$$

D_{50} , llamado definición, es la relación de energía dentro de los primeros 50 ms frente a la energía general del sonido. Suele usarse para definir las condiciones del habla. Junto con el C_{80} , proporciona el equilibrio entre el arribo de energía temprana y tardía [7]. Se representa con la siguiente ecuación:

$$D_{50} = \frac{\int_{0ms}^{50ms} p^2(t) dt}{\int_{50ms}^{\infty} p^2(t) dt} \quad (11)$$

2.6. RUIDO DE FONDO

Se define como ruido de fondo al nivel de sonido en un lugar y tiempo determinados, medido en ausencia de cualquier otro sonido intermitente. También se lo conoce como ruido ambiental o residual [8]. En nuestro trabajo, será cualquier fuente sonora que no sea necesaria para la medición.

2.7. SUAVIZADO

La técnica de la Transformada de Hilbert consiste en un filtro de una señal LTI dado por la convolución entre la misma y $\frac{1}{\pi t}$. Se utiliza para obtener la envolvente de la señal [9]. Está representada por la siguiente ecuación:

$$H[g(t)] = g(t) * \frac{1}{\pi t} \quad (12)$$

Siendo $g(t)$ la señal a transformar.

La técnica del filtro de media móvil opera tomando una cantidad de valores alrededor de una muestra de la señal de entrada y devolviendo su promedio en la salida. Al promediar los valores de forma local, las componentes rápidas de alta frecuencia serán promediadas y las variaciones mas lentas de frecuencia serán mantenidas, lo cual corresponde a suavizar o aplicar un filtro pasa bajos [10].



La técnica del filtro Savitzky-Golay se relaciona con ciertas cosas del de media móvil. Lo que hace es tomar ventanas con un número impar de muestras, y se obtiene un polinomio de grado n que mejor aproxima la señal a esa ventana. Permite elegir el grado del polinomio y la longitud de la ventana [11]



También se usa la Integral de Schroeder para generar la curva de decrecimiento de la respuesta al impulso suavizada, y así poder tener una señal mas adecuada para calcular los parámetros [3]. Para eso se utiliza la siguiente ecuación:

$$E(t) = \int_t^{\infty} p^2(\tau) d\tau = \int_0^{\infty} p^2(\tau) d\tau - \int_0^t p^2(\tau) d\tau \quad (13)$$

Siendo p la respuesta al impulso y $E(t)$ la energía de la curva de decrecimiento en función del tiempo.

3. DESARROLLO EXPERIMENTAL

El trabajo abarca las siguientes etapas.

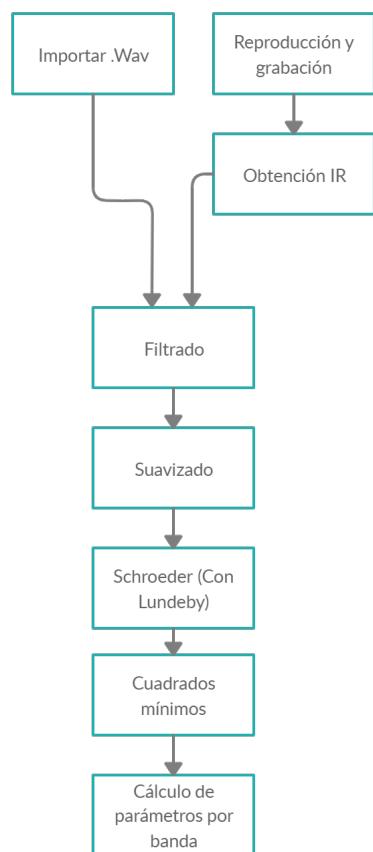


Figura 4: Diagrama de flujo de las etapas del trabajo.

Para cada una de las etapas se define una función, la cual recibe los necesarios parámetros de entrada y devuelve los requeridos parámetros de salida. Mediante estos últimos, cada función le proporciona los datos a la siguiente.

3.1. ADQUISICIÓN

El software cuenta con la posibilidad de generación, grabación, reproducción y carga de archivos. En primer instancia se sintetiza un ruido rosa que se utiliza para ajustar el nivel de la fuente al menos a 45dB por encima del nivel de ruido de fondo como especifica la norma ISO 3382 [3]. Con la función desarrollada `pinkNoise` se sintetiza este ruido en dos pasos. Primero, se genera un vector de valores aleatorios con la cantidad de muestras deseadas utilizando Numpy. Y luego, a dicho vector se le aplica un filtro utilizando métodos de la librería `Scipy.signal` que genera la pendiente de -3dB por octava. Luego se genera un sine sweep logarítmico mediante la ecuación [1] y su respectivo filtro inverso [4]. Para lograr esto se define la función `sineSweep` que toma como variables de entrada los valores tiempo y frecuencias para el barrido. Esta

función devuelve una matriz de Numpy que contiene los valores de amplitud para cada muestra del sine sweep.

El sine sweep logarítmico generado se reproduce en el recinto y se lleva a cabo la grabación del mismo usando en Python la librería `Sounddevice`. También está la posibilidad de cargar archivos de audio mediante la función `importWavs`, la cual utiliza métodos de la librería `Soundfile`. Esto se utiliza para obtener los parámetros solicitados de mediciones preexistentes.

En el caso de que no se cuente con la posibilidad de cargar archivos o de grabarlos, el programa permite sintetizar una respuesta al impulso. Lo cual ofrece una prueba de las capacidades del programa. Se genera a partir de valores predeterminados de T60, especificando el ancho de banda deseado (octava o tercio de octava).

Para la obtención de la respuesta al impulso se define una función `iRObtention` a partir del método propuesto por Farina [4], la cual toma como parámetros de entrada los Numpy array de la grabación del sine sweep logarítmico y su respectivo filtro inverso. Luego se convoluciona los mismos utilizando una FFT de la misma librería que da como resultado la respuesta al impulso del recinto.

3.2. PROCESAMIENTO

En la primer etapa de procesamiento se define la función `filtr` para filtrar la respuesta al impulso de la señal por octavas y por tercios de octava, utilizando los criterios de filtrado de la norma IEC 61260 [2]. Esto se hace para obtener los parámetros acústicos deseados por bandas de frecuencias. La misma recibe como parámetros de entrada el Numpy array de la respuesta al impulso, la frecuencia de muestreo y una cadena de caracteres que permite elegir el ancho de banda deseado ('octave' o 'third').

La señal luego pasa por un proceso de suavizado, dado por la aplicación de las técnicas de la Transformada de Hilbert, filtro de media móvil y filtro de Savitzky-Golay. Para la implementación de estas se define la función `Smoothing` que recibe como parámetros de entrada el Numpy array de la respuesta al impulso, la frecuencia de muestreo y de ser necesario, el tamaño de ventana y el grado del polinomio. Se utilizan las respectivas funciones de los filtros de la librería `Scipy.Signal`.

En las figuras 5, 6 y 7 se puede apreciar en azul una señal de ejemplo y en rojo cómo actúa el respectivo suavizado. Se representa la amplitud en función del tiempo.

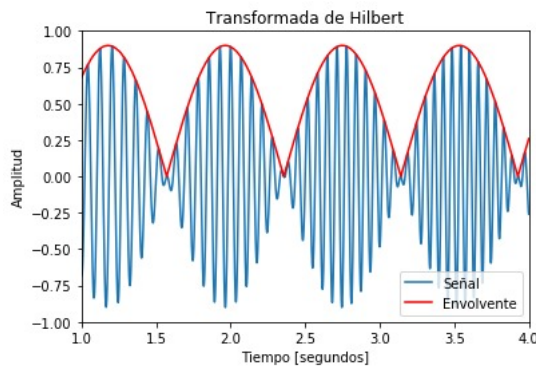


Figura 5: Gráfico transformada Hilbert.

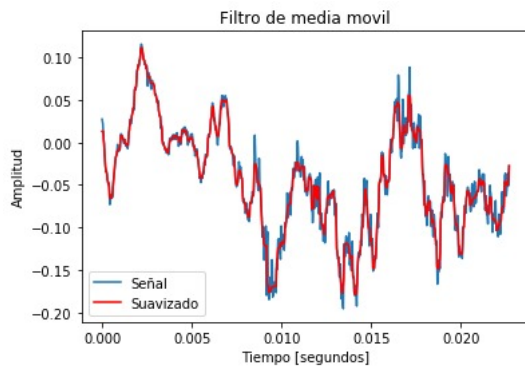


Figura 6: Gráfico Filtro de promedio móvil.

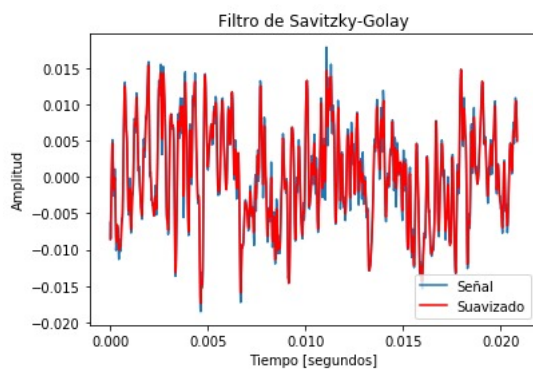


Figura 7: Gráfico filtro Savitzky-Golay.

Luego utilizamos el método de Lundebj para seleccionar el intervalo temporal más adecuado como límite superior de la integral de Schroeder, buscando eliminar contribuciones no deseadas.

Luego se aplica la Integral de Schroeder al Numpy array mediante la ecuación (13).

Se define la función `leastSquares`, la cual permite aplicar una regresión lineal por cuadrados mínimos al Numpy array contenido. Se aplica mediante la función `polyfit` ofrecida también por Numpy.

Para calcular los parámetros acústicos, se definen funciones por cada parámetro, que parten de un array de Numpy que contiene la recta aproximada de la respuesta al impulso suavizada. El EDT se calcula ajustando con `polyfit` una nueva recta entre los valores de -1dB y -10dB. A partir de ese ajuste, se eliminan todos los valores que sean menores a -10dB en el array y se toma la cantidad de muestras restantes, las cuales, al dividir por la frecuencia de muestreo, dan el valor de EDT en segundos.

El método es el mismo para calcular el valor de T_{60} a partir de los valores de T_{10} , T_{20} y T_{30} , ajustando la nueva recta a partir de la caída que representa cada parámetro. Al final, si se obtuvo el valor del T_{10} , se lo multiplica por 6 para obtener el T_{60} , y así con cada uno. El usuario tiene la posibilidad de elegir a partir de qué parámetro calcularlo.

Para calcular los valores de C80 y D50 se aplican las ecuaciones (10) y (11) al array de Numpy, utilizando las funciones matemáticas requeridas de la misma librería.

3.3. Implementación Web

Se decide realizar como interfaz gráfica una página web, la cual cuenta con las secciones necesarias para hacer el código y su respectiva utilización funcional al usuario.

La construcción de la página se realiza a partir de la librería Flask de Python (backend), HTML5 y CSS3 (frontend). Dicha librería cuenta con métodos sencillos para implementar el ruteo y la adquisición de datos seleccionados por el usuario a partir de formularios.

El sitio consta de tres ramas principales con métodos de los cuales partir; Obtención a partir del `sinesweep` generado (/), carga (/upload) y síntesis de respuesta al impulso (/synthesize). Esta última se utiliza a modo de prueba de la aplicación.

Una vez obtenida, se avanza a la sección de procesamiento (/process) donde se selecciona el tipo de filtro (octava, tercio de octava), a partir del cual se calcularán los parámetros posteriormente, y el método con el cual calcular el valor de T_{60} (t_{10} , t_{20} , t_{30} (ver 3.2)).

A partir de esto, se lleva al usuario a la sección de resultados (/results) donde puede ver una tabla con los parámetros por división de banda frecuencial seleccionada y un gráfico de la respectiva respuesta al impulso. Además, el usuario cuenta con la opción de descargar un archivo .txt con los vectores de dichos parámetros separados por las bandas elegidas.

4. RESULTADOS

Se prueba el software desarrollado reproduciendo el sine sweep logarítmico en una habitación y midiendo con un micrófono de medición. Se comparan los resultados obtenidos con los proporcionados a partir de la misma grabación por el módulo Aurora en el DAW Audacity a modo de validación de resultados. La primera diferencia dentro de los resultados parciales en el cálculo de los parámetros se da en la obtención de la respuesta al impulso a partir de la convolución. Si se analiza dentro de la teoría desarrollada en este trabajo, el resultado concuerda con lo esperado. Sin embargo, Aurora devuelve un resultado que difiere con otra pendiente y por ende tiempo de reverberación, visto en la Figura 8. Para ahondar en esta diferencia, se debería comparar dentro del código fuente del módulo Aurora si los cálculos son los mismos para poder dar una mejor descripción de la comparación.

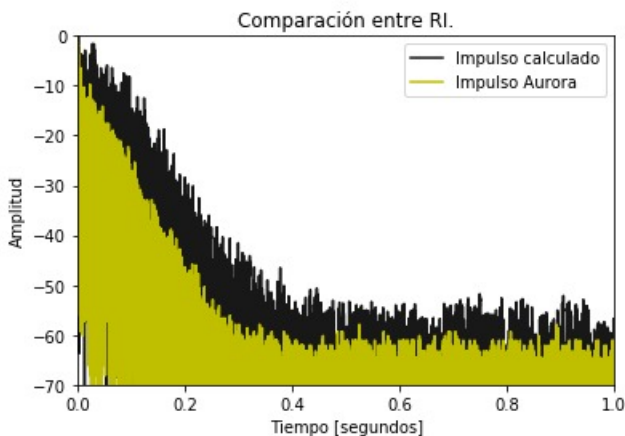


Figura 8: Comparación de la respuesta al impulso calculada por nuestro software y por Aurora.

Luego se realiza la comparación entre los resultados de los descriptores acústicos C80, D50, EDT y T60, vistos en las Tablas 9 y 10. En general los resultados son similares, a excepción de la banda de 125Hz. Esta diferencia se podría pensar debido al flujo dado por las funciones en el software diseñado. Ya que este trata a todas las bandas de frecuencia de la misma forma, y los ajustes realizados por Aurora pueden cambiar sus parámetros internos para cada banda de frecuencia. Resultaría ideal por ende, revisar los parámetros de las funciones de lundeby, suavizados y demás para cada una de las bandas, sobretodo, para bajas frecuencias.

Frq.band [Hz]	125	250	500	1k	2k	4k
C80 [dB]	17.0	15.0	15.9	15.9	16.0	14.3
D50 [%]	73.2	118.0	89.2	86.3	85.2	124.0
EDT [s]	0,266	0,241	0,261	0,265	0,269	0,231
T60 [s]	0.619	0.534	0.576	0.584	0.595	0.518

Figura 9: Tabla de los parámetros calculados con el software diseñado.

Frq.band [Hz]	125	250	500	1k	2k	4k
C80 [dB]	12,216	18,334	18,51	17,652	18,853	20,199
D50 [%]	88,759	95,819	93,689	94,144	94,772	96,032
EDT [s]	0,421	0,305	0,324	0,289	0,268	0,27
T60 [s]	0,842	0,61	0,648	0,578	0,536	0,54

Figura 10: Tabla de los parámetros calculados con el módulo Aurora.

5. Conclusiones

Las conclusiones que se pueden extraer del trabajo realizado son las siguientes.

En primer lugar, que la implementación del lenguaje Python 3 resultó satisfactoria y que, con más tiempo, se podría desarrollar un producto estable para distribuir entre profesionales y técnicos de las ciencias acústicas para calcular de modo semiautomático los descriptores acústicos de una sala.

En segundo lugar, las diferencias entre el módulo Aurora y el software desarrollado son mínimas pero implican que hay errores por descubrir aun dentro del código fuente. Y, por lo tanto, un flujo de trabajo que se podría mejorar.

Finalmente, y a partir de la última entrega, se decide continuar con el desarrollo del software para lograr un producto final y abierto al público.

Referencias

- [1] L. Musolino. *Desarrollo de software para el cálculo de parámetros acústicos bajo norma ISO 3382*. Señales y Sistemas, Ingeniería de sonido, Universidad Nacional de Tres de Febrero, 2019.
- [2] IEC-International Electrotechnical Commission et al. Iec 61260-3. 2016.
- [3] Iso 3382:1997. acoustics - measurement of room, acoustic parameters. Standard, March 1997.
- [4] A. Farina. *Simultaneous Measurement of Impulse Response and Distortion With a Swept-Sine Technique*. Dipartimento di Ingegneria Industriale, Università di Parma., 2000.
- [5] A. Domingo. *Apuntes de Acústica*. Creative Commons., 2014.

- [6] I. Miranda A. Millan, D. Galvan. *Diseño e implementación de prototipo funcional de reverberador por convolución en tiempo real*. Programa de Ingeniería de Sonido, Facultad de Ingeniería, Universidad de San Buenaventura., 2008.
- [7] S. Vorländer M Havelock, D. Kuwano. *Handbook of Signal Processing in Acoustics*. Springer Science Business Media, 2008.
- [8] *Sound and Vibration Instrument Hire and Calibration Background Noise Measurement*. Gracey Associates.
- [9] F. Kschischang. *The Hilbert Transform*. The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, 2006.
- [10] A Oppenheim. *Señales y Sistemas*. Pearson Education, 1998.
- [11] R Gonzalez. *Evaluación de logaritmos para estimar la derivada de presión en pruebas de pozos*. Universidad Surcolombiana, Facultad de Ingeniería.