



# Informe sobre el desarrollo de software para el cálculo de parámetros acústicos según norma ISO 3382

Pedro Manuel Díaz Varela

Señales & Sistemas, Ingeniería de sonido  
Universidad Nacional de Tres de Febrero , Buenos aires, Argentina.  
pedro.divar@gmail.com

## Resumen

En el presente trabajo se realizó un informe **se** explica el desarrollo de software para el cálculo de parámetros acústicos según la norma ISO 3382[5]. El desarrollo consiste en una serie de funciones implementadas en MatLab las cuales **se** permiten grabar y reproducir en simultaneo un sine-sweep, obtener respuestas al impulso, reproducir ruido rosa **se** y procesar señales para la búsqueda de parámetros acústicos. Si bien los resultados se ven insatisfactorios este trabajo presenta un buen punto de partida para estudiar los métodos para la obtención de parámetros acústicos.

**Keywords:** ISO 3382, IR

## 1. INTRODUCCIÓN

En este informe se tienen como objetivo la descripción de las funciones implementadas en el lenguaje de programación MatLab para el cálculo de parámetros acústicos según la norma ISO 3382:1997[5]. Dichas funciones permiten obtener una respuesta al impulso mediante la reproducción y grabación en simultaneo de un sine-sweep en un **resinto**. También se implementó una serie de filtros para el análisis de las señales por frecuencia y una función que sintetiza una respuesta al impulso. Por otra parte, si se desea calibrar la fuente usada para la medición, se encuentra disponible una función que genera un ruido rosa acorde a la normativa ISO 3382

be la alternativa de obtener el T60 a partir de la extrapolación al decaimiento temprano de la señal ya sean caídas de 10 [dB](T10), 20 [dB](T20), o 30 [dB](T30). También se estudian otros parámetros como el EDT(early decay time) definido como el tiempo que tarda la señal en decaer 10[dB] multiplicado por 6 y la definición (D50) y la claridad(C80), estos dos últimos parámetros se definen en la norma ISO 3382[5].

### 2.2. Ruido Rosa

## 2. MARCO TEÓRICO

### 2.1. Tiempos de reverberación

Este trabajo se centra en la obtención de un parámetro acústico denominado T60. Este parámetro se define como el tiempo requerido para que un nivel de presión sonora (SPL) decaiga 60 [dB] luego de que la fuente sonora deje de actuar según la norma ISO 3382[5]. Para conseguir este SPL se requiere que el rango dinámico de la medición sea de 75 [dB], pero la normativa ISO 354:2003[1] descri-

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. El nivel del ruido rosa por intervalo fijo de frecuencias va decayendo a razón de 3 dB/octava de forma no lineal como se ve en la *Figura 1*. Esta señal es de interés para realizar un ajuste, de la ganancia de la fuente que luego reproducirá el sine-sweep al realizar una medición, teniendo en cuenta que el ruido de fondo no supere los 45[dB].

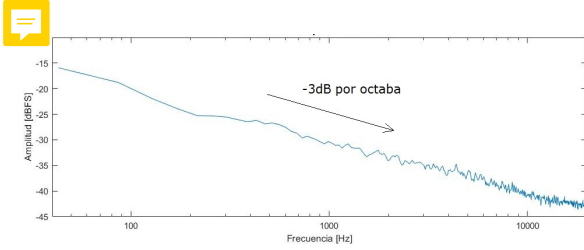


Figura 1: Respuesta en frecuencia de un ruido rosa de 1 segundo generado por código.

### 2.3. Sine-sweep y filtro inverso

El sine-sweep, al ser un barrido de frecuencias en el tiempo, permite concentrar la energía en una frecuencia específica en cada momento, mejorando la relación señal-ruido ~~a comparación~~. Debido a la duración, distribución espectral y baja relación señal-ruido de una señal impulsiva generada por una explosión, el sine sweep es el método más indicado para generar el impulso por medio de la convolución [4]. El sine sweep logarítmico vendrá dado por la siguiente expresión:

$$x(t) = \sin[\theta(t)] = \sin\left[K\left(e^{\frac{t}{L}} - 1\right)\right] \quad (1)$$

Donde:

$$K = \frac{T\omega_1}{\ln\left(\frac{\omega_2}{\omega_1}\right)} \quad (2)$$

y

$$L = \frac{T}{\ln\left(\frac{\omega_2}{\omega_1}\right)} \quad (3)$$

Siendo  $\omega_1$  y  $\omega_2$  las frecuencias ~~as~~ angulares inferior y superior respectivamente, y el tiempo de duración del sine sweep en segundos. Como se puede observar en la *Figura 2* el espectro de la señal no es plano, sino que disminuye a razón de -3 dB/octava,

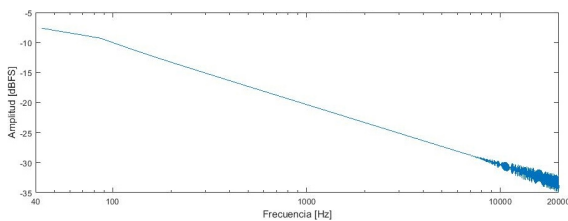


Figura 2: Respuesta en frecuencia de un sine-sweep generado por código.

Para obtener el filtro inverso correspondiente al sine-sweep, en primer lugar hay que invertirlo en el tiempo, obteniendo  $x(-t)$ . A su vez es necesaria una modulación en amplitud que compense la diferencia de energía respecto de la señal de entrada original. Luego se define matemáticamente como:

$$\omega(t) = \frac{d\theta(t)}{dt} = \frac{K}{L} e^{\frac{t}{L}} \quad (4)$$

Donde:

$$m(t) = \frac{\omega_1}{2\pi\omega(t)} \quad (5)$$

$$k(t) = m(t)x(-t) \quad (6)$$

### 2.4. Respuesta al impulso

La respuesta al impulso se define como la presión sonora en un punto del resinto en estudio como resultado de la emisión de un impulso efectuado en otro punto. En este trabajo se evalúa la respuesta al impulso convolucionando un sine-sweep grabado con su filtro inverso utilizando la transformada rápida de Fourier.

### 2.5. Suavizado

Se utilizan en el presente trabajo dos técnicas de suavizado. La primera se basa en aplicar la transformada de Hilbert y tomar su valor absoluto. Por otro lado se utiliza la integración hacia atrás de Schroeder la cual permite producir una curva de caída monótona y suavizada. Si definimos un impulso como  $h(t)$  sobre el intervalo de medición  $[0, T]$  la expresión de dicha integración viene dada por la integral:

$$y(t) = \frac{\int_t^T h(t)^2 dt}{\int_0^T h(t)^2 dt} \quad (7)$$

Pero la integral de Schroeder presenta un problema cuando el piso de ruido de fondo se incluye dentro del intervalo de integración. Existen diferentes métodos para suprimir los efectos del ruido de fondo, en nuestro caso utilizamos el método de Lundebly[3] el cual determina automáticamente el nivel de ruido de fondo, el punto de truncamiento del ruido de decaimiento y la pendiente de decaimiento tardía de una respuesta de impulso. Este método actúa a partir del promedio cuadrático de la Iren intervalos locales donde se estima el nivel inicial de ruido de fondo en la parte tardía de la señal. Luego a ~~partir~~ de la pendiente de decaimiento que se obtiene mediante una regresión lineal, en el intervalo temporal correspondiente, se determina el punto de truncamiento inicial como la intersección de la pendiente de decaimiento y el ~~nivel~~ de ruido de fondo. A partir de lo anterior se ~~determinan~~ nuevos ~~intervalos temporales~~ y una nueva pendiente de caída tardía. El proceso se repite iterativamente hasta un máximo de 5 iteraciones.

### 3. DESARROLLO EXPERIMENTAL

En el desarrollo experimental se utilizaron funciones para encapsular cada problema y luego poder utilizar y testear cada una de estas funciones por separado. Las funciones principales implementadas fueron:

- sineSweep
- grabarReproducir
- ruidoRosa
- respuestaAlImpulso
- repuestaSintetizada
- filtradoPorTercioOctavas
- filtradoPorOctavas
- suavizado
- calcularTiempos
- calcularRI
- gui

#### 3.1. sineSweep

La función sineSweep se encarga de generar dos archivos .wav con un sine-sweep y su filtro inverso que luego serán utilizados por la función grabarReproducir. Se utilizó la función plotAudio para graficar y verificar el resultado de esta función como se puede ver en la *Figura 3* y la *Figura 4*.

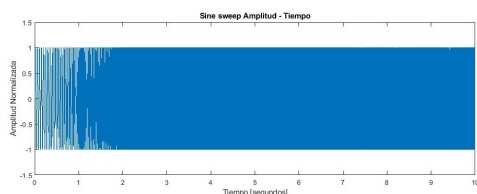


Figura 3: Sine sweep en función del tiempo.

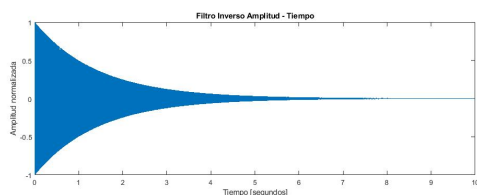


Figura 4: Filtro inverso en función del tiempo.

#### 3.2. grabarReproducir

En orden de obtener los datos en tiempo real a lo largo de las mediciones, utilizamos la función grabarReproducir que se vale de la función audiorecorder y getaudiodata para crear un vector con los valores obtenidos en tiempo. La función audiorecorder genera un objeto de audio con valores predeterminados de frecuencia de muestreo de 8000 muestras por segundos, con una profundidad de bits de 8, lo cual se ha modificado para que sea un valor que determine el usuario de la función. Este objeto nos ayuda a grabar asincrónicamente con la función record la cual será ejecutada en paralelo con sound como se ve en la *Figura 5*. La función pause determina en segundos la interrupción de la grabación realizada, se consideraron 5 segundos antes de empezar a grabar debido a que es necesario que se grabe sin que hayan personas en el lugar de la medición y que la grabación dure el doble de lo que dura el sine sweep ya que el sistema a medir tiene que volver a su estado de equilibrio cuando no hay entrada. Luego se consideró dejar 1 segundo grabando antes de reproducir el sine-sweep con la función sound para no perder muestras. Como podemos observar en la *Figura 5* la grabación y el filtro inverso usados son finalmente guardados como un archivo .wav en la carpeta mediciones.

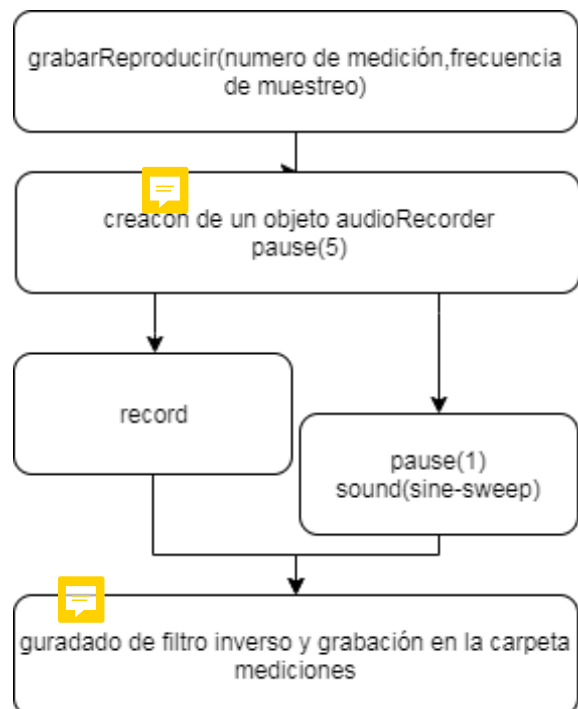


Figura 5: Diagrama de flujo de la función grabarReproducir.

### 3.3. respuestaAlImpulso y respuesta-Sintetizada

La función `respuestaAlImpulso` toma dos vectores que luego convolucionara utilizando la transformada rápida de fourier, luego recortará la funcion en la zona donde nos intereza analizarla y finalmente guardará el impulso en una carpeta a elección del usuario. En orden de visualizar un impulso sintetico se utilizó la función `respuestaSintetizada` que tendrá como parámetros las bandas centrales, el T60 y la frecuencia de muestreo. También guardará el impulso sintetizado en una carpeta a elección del usuario..

### 3.4. filtradoPorTercioOctavas y filtradoPorOctavas

Se programó dos funciones de filtrado que al pasar como parámetro una señal filtran la señal por bandas de octava y tercio de octava con la ayuda de la función `designfilt` con un orden de 6 y 8 respectivamente, estos filtros son filtros pasa banda y respetan la norma IEC 61260[2]. La salida de estas funciones son un conjunto de señales agrupadas en un cellarray filtradas en cada una de las bandas en las cuales se filtró, además devuelve un struct con los filtros diseñados. Se utilizó la función `fvtool` para graficar los filtros diseñados como se observa en la *Figura 6* y en la *Figura 7*.

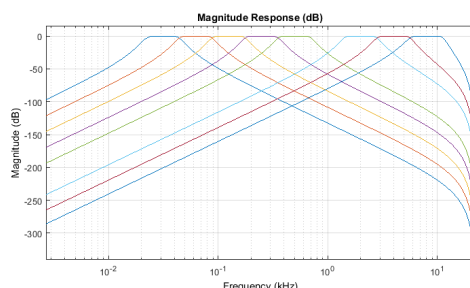


Figura 6: Filtros de octava.

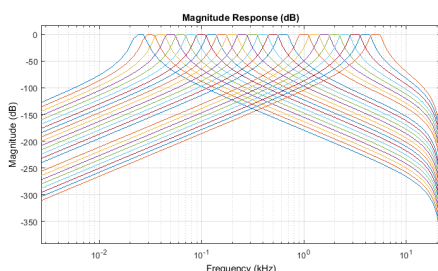


Figura 7: Filtros de octava.

### 3.5. suavizado

La función de suavizado utiliza el absoluto de la transformada de una señal y luego calcula el extremo de integración de Schroeder con el metodo Lundebey explicado previamente para luego efectuar la integración de Schroeder,obteniendo así la envolvente de la señal y finalmete aplicarle un logaritmo para que quede una señal de energía.

### 3.6. calcularTiempos

Para la obtención de los descriptores se realizan regresiones por el método de cuadrados mínimos. El EDT es calculado tomando el tiempo necesario para que la recta de aproximación decaiga desde 0 [dB] hasta -10 [dB], para el T10 el tiempo necesario para que la recta decaiga de -5[dB] a -15[dB],para el T20 el tiempo necesario para que la recta decaiga de -5[dB] a -25[dB], para el T30 el tiempo necesario para que la recta decaiga de -5[dB] a -35[dB]. Finalmente se calculan los parametros C80 y D50.

### 3.7. calcularRI

A partir de un impulso pasado como parámetro está función te devuelve dos tablas con los parametros acústicos anteriormente mencionados en diferentes salas. Está función ~~se primero~~ filtra la señal luego suaviza y calcula los parámetros para cada banda filtrada.

### 3.8. gui

Se quiso implemetar una interfaz en la que se desarrollen las funciones ~~y hay facilidad~~ de manejo por parte de usuarios que no conocen la implementación de las funciones. Para esto se pensó organizar la interfaz en pestañas con diferentes botones y popups y ir modificando los datos y guardando tablas dinámicamente sin previo conocimiento de MATLAB. Debido a problemas de tiempo la interfaz quedó sin terminar ya que faltaron implementar varios callbacks, actualizar los popups según iba cambiando la base de datos y presentar los datos con los valores de los resultados.

### 3.9. Dataset

Para validar las funciones implentadas se utilizaron un conjunto de respuestas al impulso de 3 recintos diferentes:

- Iglesia de San Nicola
- Sala de tipadp de una fabrica de chocolate de Nueva York
- Sala de conferencia el Octagono [6]

Las mediciones de la fábrica y la sala de conferencias fueron echas con el método de sine sweep mientras que la medición de la iglesia fue echa con el impulso generado de el estallido de un globo. Como se puede obserbar en la figura 10 en el octagono se realizaron 144 mediciones de las cuales se utilizaron 4 en las posiciones:

- $x=4,y=4$
- $x=4,y=8$
- $x=8,y=4$
- $x=8,y=8$



Figura 8: Iglesia de San Nicola.

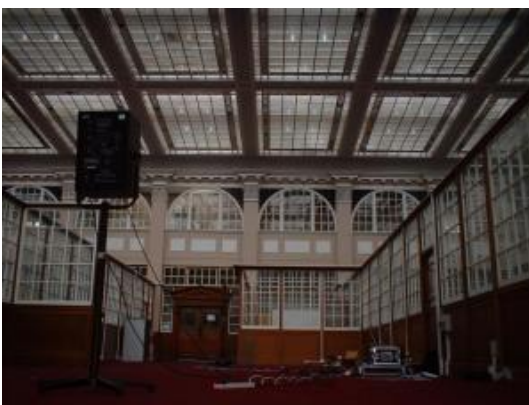


Figura 9: Sala de tipadp de una fabrica de chocolate de Nueva York.

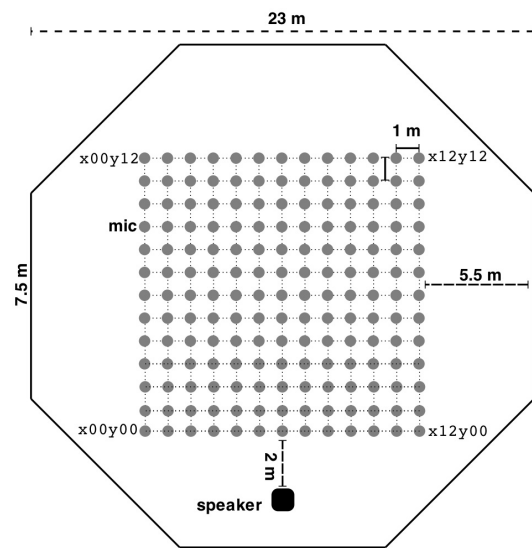


Figura 10: Sala de conferencia el Octagono.

## 4. RESULTADOS

Se utilizó el software Audacity y el plugin Aurora como método de validación de los resultados. Como se puede obserbar en las tablas a continuación las bandas analizadas fueron las de las frecuencias centrales 125, 500, 1000, 2000 y 4000 hz.

Tabla 1: Decriptores calculados con software propio

- Toma x04y04

<i>Descriptor</i>	<i>125Hz</i>	<i>250Hz</i>	<i>500Hz</i>	<i>1000Hz</i>	<i>2000Hz</i>	<i>4000Hz</i>
EDT[s]	0,3454	2,2794	3,0981	3,2051	2,8265	2,3061
T10[s]	0,2782	2,3351	2,8153	3,0230	2,7244	2,3510
T20[s]	0,2527	2,2290	2,6614	2,8299	2,5832	2,3018
T30[s]	0,3429	2,3882	3,0120	3,4462	2,8734	2,3658
D50[porc.]	26,8849	31,9489	61,7156	54,5630	65,5761	72,5664
C80[dB]	-3,6152	-2,1516	2,5650	1,0217	2,9771	4,3764

Tabla 2: Decriptores calculados con software propio

- Toma x04y08

<i>Descriptor</i>	<i>125Hz</i>	<i>250Hz</i>	<i>500Hz</i>	<i>1000Hz</i>	<i>2000Hz</i>	<i>4000Hz</i>
EDT[s]	1,7729	2,4083	2,9304	3,1299	2,8358	2,3463
T10[s]	1,6059	2,2476	2,8416	2,9992	2,7412	2,3655
T20[s]	1,3976	2,1923	2,6839	2,7758	2,6055	2,3099
T30[s]	2,0689	2,4301	3,1178	3,1926	2,9148	2,3907
D50[porc.]	19,4659	14,6373	18,8125	28,8570	46,4350	51,8259
C80[dB]	-3,8666	-4,1114	-4,5558	-3,0446	0,4190	1,1587

Tabla 3: Decriptores calculados con software propio

- Toma x08y04

<i>Descriptor</i>	<i>125Hz</i>	<i>250Hz</i>	<i>500Hz</i>	<i>1000Hz</i>	<i>2000Hz</i>	<i>4000Hz</i>
EDT[s]	0,0100	2,3744	3,0282	3,2122	2,8020	2,3544
T10[s]	0,0080	2,4573	2,8977	2,9796	2,7473	2,3567
T20[s]	0,0072	2,3343	2,6828	2,7699	2,5925	2,2962
T30[s]	0,0207	2,4209	2,9945	3,3084	2,8225	2,4145
D50[porc.]	26,1838	29,2850	60,3696	53,8928	60,8649	71,8087
C80[dB]	-4,2977	-3,2359	2,1703	1,0345	2,05920	4,1759

Tabla 4: Decriptores calculados con software propio

- Toma x08y08

<i>Descriptor</i>	<i>125Hz</i>	<i>250Hz</i>	<i>500Hz</i>	<i>1000Hz</i>	<i>2000Hz</i>	<i>4000Hz</i>
EDT[s]	1,7630	2,3297	3,2536	3,2014	2,7634	2,2786
T10[s]	1,3373	2,3566	2,9034	3,0801	2,7359	2,3441
T20[s]	1,1987	2,2224	2,7863	2,8725	2,6106	2,2975
T30[s]	2,0740	2,2756	3,3236	3,4002	2,8059	2,3548
D50[porc.]	16,6318	15,1860	21,6265	28,7541	44,0506	53,3864
C80[dB]	-4,9849	-4,7259	-4,1991	-3,1136	-0,3382	1,1954

Tabla 5: Decriptores calculados con Aurora - Toma  
x04y04

<i>Descriptor</i>	<i>125Hz</i>	<i>250Hz</i>	<i>500Hz</i>	<i>1000Hz</i>	<i>2000Hz</i>	<i>4000Hz</i>
EDT[s]	1,569	2,4104	3,0300	3,5601	2,7419	2,1523
T20[s]	.-	2,2804	3,0303	3,2650	2,8769	2,2749
T30[s]	.-	2,1285	2,8799	3,0046	2,8174	2,3708
D50[porc.]	44,7444	38,5152	62,2214	54,5593	66,3621	73,1248
C80[dB]	0,7847	-1,0251	2,6982	1,5071	3,6502	5,3449

Tabla 6: Decriptores calculados con Aurora - Toma  
x04y08

<i>Descriptor</i>	<i>125Hz</i>	<i>250Hz</i>	<i>500Hz</i>	<i>1000Hz</i>	<i>2000Hz</i>	<i>4000Hz</i>
EDT[s]	0,7984	2,4335	3,2066	3,1698	2,8267	2,1651
T20[s]	0,2052	2,1878	2,8976	3,2194	2,8816	2,3628
T30[s]	0,1988	1,9016	2,7221	2,9250	2,8162	2,3940
D50[porc.]	59,0070	28,3990	31,3163	37,0879	52,5973	58,3939
C80[dB]	4,7828	-1,1681	-1,4190	-0,8534	2,8254	4,0466

Tabla 7: Decriptores calculados con Aurora - Toma  
x08y04

<i>Descriptor</i>	<i>125Hz</i>	<i>250Hz</i>	<i>500Hz</i>	<i>1000Hz</i>	<i>2000Hz</i>	<i>4000Hz</i>
EDT[s]	7,0353	2,3990	2,9725	3,3523	2,2845	1,9577
T20[s]	6,6478	2,5527	3,1653	3,2169	2,9120	2,3055
T30[s]	6,2447	2,4499	2,9284	3,0237	2,8792	2,3643
D50[porc.]	34,6937	32,4235	60,4253	54,5440	61,8871	72,2484
C80[dB]	-2,2018	-2,2081	2,4547	1,6186	3,4383	5,6084

Tabla 8: Decriptores calculados con Aurora - Toma  
x08y08

<i>Descriptor</i>	<i>125Hz</i>	<i>250Hz</i>	<i>500Hz</i>	<i>1000Hz</i>	<i>2000Hz</i>	<i>4000Hz</i>
EDT[s]	0,8215	2,0376	3,3795	3,4760	2,6427	2,0916
T20[s]	.-	2,2948	2,9581	3,2901	2,8517	2,2889
T30[s]	.-	1,9776	2,8018	3,0168	2,8023	2,3491
D50[porc.]	47,8508	24,5569	28,7239	34,7100	50,2245	58,3290
C80[dB]	2,3024	-2,1110	-1,5878	-0,2791	2,2584	3,6329

Tabla 9: Decriptores calculados con software propio

- Fabrica de chocolate

<i>Descriptor</i>	<i>125Hz</i>	<i>250Hz</i>	<i>500Hz</i>	<i>1000Hz</i>	<i>2000Hz</i>	<i>4000Hz</i>
EDT[s]	1,2821	2,1179	2,7923	2,6757	2,0721	1,4669
T10[s]	1,4146	2,1272	2,9332	2,7645	2,1622	1,5447
T20[s]	1,5441	2,1586	2,9469	2,7827	2,2133	1,5902
T30[s]	0,9740	1,8157	2,6226	2,4243	1,9211	1,3853
D50[porc.]	40,7143	27,4203	40,8012	27,7342	59,2350	52,7987
C80[dB]	2,6683	-0,0093	0,1502	-1,3080	3,5488	4,0651

Tabla 10: Decriptores calculados con software propio - Iglesia

<i>Descriptor</i>	<i>125Hz</i>	<i>250Hz</i>	<i>500Hz</i>	<i>1000Hz</i>	<i>2000Hz</i>	<i>4000Hz</i>
EDT[s]	1,3108	0,9095	0,9163	0,9634	0,7537	0,6815
T10[s]	1,1670	0,8433	0,9128	0,9408	0,7983	0,7211
T20[s]	1,1460	0,9040	0,9325	0,9751	0,8182	0,7261
T30[s]	0,8363	0,9349	0,8857	0,8621	0,7421	0,6181
D50[porc.]	79,2874	58,8156	74,2799	74,2957	71,5339	75,0721
C80[dB]	9,6740	4,1406	7,3237	7,6109	6,9644	9,5333

Tabla 11: Decriptores calculados con Aurora - Fabrica de chocolate

<i>Descriptor</i>	<i>125Hz</i>	<i>250Hz</i>	<i>500Hz</i>	<i>1000Hz</i>	<i>2000Hz</i>	<i>4000Hz</i>
EDT[s]	0,8889	1,5588	2,3862	2,1842	1,6140	1,1560
T20[s]	1,2818	2,1187	2,9227	2,7421	2,0602	1,4625
T30[s]	1,6043	2,1850	2,9946	2,8091	2,1544	1,5467
D50[porc.]	62,8749	47,4576	50,1803	40,8934	67,6588	69,7009
C80[dB]	4,3934	2,5818	1,0498	-0,2428	4,1308	5,6559

Tabla 12: Decriptores calculados con Aurora - Iglesia

<i>Descriptor</i>	<i>125Hz</i>	<i>250Hz</i>	<i>500Hz</i>	<i>1000Hz</i>	<i>2000Hz</i>	<i>4000Hz</i>
EDT[s]	0,7764	1,0762	0,8741	0,7344	0,7199	0,5509
T20[s]	1,2079	0,8177	0,9040	0,9461	0,7728	0,7022
T30[s]	1,1476	0,8729	0,9546	0,9726	0,8138	0,7397
D50[porc.]	56,8725	45,4893	71,2081	72,6654	71,0181	74,8106
C80[dB]	5,9718	2,0342	6,6821	7,3368	6,9444	9,5251



## 5. CONCLUSIONES

Los resultados obtenidos a través del diseño de software para el cálculo de los tiempos de reverberación demuestran que al código le faltaría un ajuste ya que la diferencia en los cálculos entre el software comercial y los datos obtenidos en nuestro código son de por lo menos 0,2 segundos y llegan a variar hasta 1 segundo. Si bien los datos obtenidos a través del aurora dan respuestas irrisorias en algunos casos (tal es el caso de la tabla 7 en la banda de los 125 Hz) los datos obtenidos no llegan a ser satisfactorios por lo que es recomendable probar con algún método de suavizado distinto ya sea aplicar media móvil o calcular el extremo de integración de Schroeder con el método Chu. También podemos concluir que los valores varían según la señal lo cual demuestra que los métodos deben ser probados ~~exhaustivamente~~ hasta encontrar resultados satisfactorios. Para el D50 podemos notar amplios márgenes de errores por lo cual deberíamos descartar el método utilizado para calcular este parámetro y utilizar algún otro. Por ~~ultimo~~ cabe destacar que el C50 presenta diferencias de no más de 10 dB con respecto a Aurora. Estos valores son despreciables para el oído humano ya que se tratan de niveles muy bajos.

## Referencias

- [1] ISO 354:2017. *Acoustics - Measurement of sound absorption in a reverberation room.*
- [2] CEI. 61260:2001. *Electroacoustics - Octave-band and fractional-octave-band filters. 2001.*
- [3] Aki Makivirta. Matti Karjalainen. *Estimation of Modal Decay Parameters from Noisy Response Measurements.*
- [4] I. Nikolić and O. H. Bjor (2003). *Building and Room Acoustics Measurements with Sine-Sweep Technique. Proceedings of the Institute of Acoustics 25.Pt 5.*
- [5] ISO. 3382:1997. *Acoustics Measurement of the reverberation time of rooms with reference to other acoustical parameters.*
- [6] Rebecca Stewart and Mark. Sandler. *Database of Omnidirectional and B-Format Impulse Responses.* in Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2010), Dallas, Texas, March 2010.