



UNIVERSIDAD DE CONCEPCIÓN  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

# FABRICACIÓN DE SONÓMETROS DE BAJO COSTO PARA MEDICIÓN Y CLASIFICACIÓN DE ENTORNOS SONOROS.

**Por: Francisco Ignacio Salinas Alarcón**

Tesis presentada a la Facultad de Ciencias Físicas y Matemáticas de la  
Universidad de Concepción para optar al grado académico de Ingeniero  
Físico.

Diciembre 2022

Concepción, Chile

**Profesor Guía: Hernán Fernando Astudillo Parra**

© 2022, Francisco Ignacio Salinas Alarcón

Ninguna parte de esta tesis puede reproducirse o transmitirse bajo ninguna forma o por ningún medio o procedimiento, sin permiso por escrito del autor.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento.

## Resumen

El crecimiento en la población, industria y urbanización han provocado un aumento en la magnitud de ruido en los entornos sonoros que nos rodean. Este es un contaminante con el que interactuamos constantemente y provoca efectos adversos en la salud de las personas y en la fauna, por ello, es de urgencia reconocer y establecer medidas que gestionen el ruido ambiental. Para la medición y registro de los niveles de ruido, se utilizan instrumentos profesionales certificados llamados sonómetros, los cuales tienen un alto valor monetario, debido a esto, en este trabajo se desarrollaron dos sonómetros de bajo costo usando ordenadores de placa simple como alternativa a estos productos, los que poseen características que no tiene un sonómetro convencional como el funcionamiento continuo, respaldo en la nube, acceso remoto, comunicación con aplicaciones web y ejecución de modelos de aprendizaje profundo para la clasificación de entornos sonoros. Se realizó una calibración ambiental a los sonómetros desarrollados con un sonómetro profesional de clase 1 y se obtuvieron resultados óptimos en las mediciones, teniendo 0,3 decibeles de diferencia en valores de promedios minutales, 95 % de exactitud en la ejecución del modelo de aprendizaje y correcto funcionamiento en sus otras características, logrando un dispositivo que reduce diez veces el costo mínimo de un sonómetro de clase 1. A través de los resultados se evidenció la capacidad de obtener instrumentos de bajo costo y de gran calidad, siendo un instrumento más accesible para empresas, facultades, estudiantes y docentes en proyectos de investigaciones, para así suplir la necesidad enfática de regularización y registros de diferentes entornos sonoros para mejorar la calidad de vida y salud de las personas.

**Keywords** – Contaminación acústica, Entorno sonoro, sonómetro, bajo costo, aprendizaje profundo

## Abstract

The growth in population, industry and urbanization have caused an increase in the magnitude of noise at the sound environments that surround us. This is a pollutant with which we constantly interact and causes adverse effects on the health of people and wildlife, therefore it is urgent to recognize and establish measures to manage environmental noise. For the measurement and recording of noise levels, certified professional instruments called sound level meters are used, which have a high monetary value, due to this, in this work two low cost sound level meters were developed using single board computers as an alternative to these products, which have features that a conventional sound level meter does not have, such as continuous operation, cloud backup, remote access, communication with web applications and execution of deep learning models for the classification of sound environments. The developed sound level meters were calibrated with a professional class 1 sound level meter and optimal results were obtained in the measurements, having 0,3 decibels of difference in values of minute averages, 95 % of accuracy in the execution of the learning model and correct operation in its other characteristics, achieving a device that reduces ten times the minimum cost of a class 1 sound level meter. Through the results, the ability to obtain low-cost and high-quality instruments was evidenced, being a more accessible instrument for companies, faculties, students and teachers in research projects, in order to meet the emphatic need for regularization and records of different sound environments to improve the environment and health of people.

**Keywords** – Noise pollution, sound environment, sound level meter, low cost, deep learning

# Índice general

<b>Resumen</b>	<b>I</b>
<b>Abstract</b>	<b>II</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Marco Teórico</b>	<b>3</b>
2.1. Onda acústica . . . . .	3
2.2. Presión sonora . . . . .	3
2.3. Nivel de presión sonora . . . . .	4
2.4. Espectro de frecuencias . . . . .	6
2.5. Transformada de Fourier . . . . .	7
2.6. Transformada discreta de Fourier . . . . .	8
2.7. Transformada rápida de Fourier . . . . .	9
2.8. Transformada de Fourier de tiempo reducido . . . . .	9
2.9. Espectrograma . . . . .	12
2.10. Bandas de Octava . . . . .	14
2.10.1. Filtro paso banda . . . . .	14
2.10.2. Relación de Octavas . . . . .	14
2.10.3. Designador de ancho de banda . . . . .	15
2.10.4. Frecuencia de referencia . . . . .	15
2.10.5. Frecuencia media exacta . . . . .	15
2.10.5.1. Frecuencia media nominal . . . . .	15
2.10.5.2. Frecuencias del borde . . . . .	15
2.10.5.3. Ancho de banda . . . . .	16
2.10.6. Banda de Octava . . . . .	16
2.10.7. Bandas fraccionarias de octava . . . . .	16
2.11. Bandas de tercios de octava . . . . .	17
2.12. Filtros de ponderación . . . . .	19
2.12.1. Filtro de ponderación Z . . . . .	20
2.12.2. Filtro de ponderación A . . . . .	20
2.13. Ponderación de tiempo . . . . .	23
2.14. Nivel de presión sonora continuo equivalente . . . . .	23
2.15. Aprendizaje automático . . . . .	24
2.15.1. Aprendizaje Supervisado . . . . .	25

2.15.2. Redes Neuronales Artificiales . . . . .	26
2.16. Perceptrón . . . . .	27
2.17. Neurona Sigmoide . . . . .	28
2.18. Función de costo . . . . .	30
2.19. Descenso de gradiente . . . . .	31
2.20. Proceso de entrenamiento de datos . . . . .	32
2.21. Aprendizaje Profundo . . . . .	33
2.21.1. Redes Neuronales Convolucionales . . . . .	33
2.21.1.1. Sobreajuste . . . . .	35
2.21.1.2. Activación ReLU . . . . .	35
2.21.1.3. Capa de agrupación . . . . .	36
2.21.1.4. Capa de abandono . . . . .	37
2.21.1.5. Agrupación media global . . . . .	38
2.21.1.6. Capa Plana . . . . .	39
2.21.1.7. Capa Densa . . . . .	39
2.21.1.8. Función SoftMax. . . . .	39
<b>3. Metodología</b>	<b>41</b>
3.1. Objetivo de medición. . . . .	43
3.2. Hardware. . . . .	43
3.2.1. Raspberry pi. . . . .	43
3.2.2. Micrófono. . . . .	44
3.2.3. Tarjeta de sonido USB. . . . .	45
3.3. Sistema Operativo. . . . .	46
3.4. Sonómetro de mano. . . . .	46
3.4.1. Hardware . . . . .	46
3.4.2. Procesamiento de datos. . . . .	47
3.5. Sonómetro Continuo. . . . .	50
3.5.1. Hardware . . . . .	51
3.5.2. Funcionamiento . . . . .	51
3.5.3. Procesamiento de datos. . . . .	51
3.5.4. Red Neuronal Profunda . . . . .	53
3.6. Homologación. . . . .	55
<b>4. Análisis</b>	<b>58</b>
4.1. Sonómetro continuo. . . . .	58
4.1.1. Montaje . . . . .	58
4.1.2. Homologación. . . . .	60
4.1.3. Visualización de datos . . . . .	64
4.1.4. Modelo de Aprendizaje profundo. . . . .	67
4.1.4.1. Visualización del modelo . . . . .	71
4.2. Sonómetro de mano. . . . .	73
<b>5. Discusión</b>	<b>81</b>
<b>6. Conclusión</b>	<b>84</b>

Índice general

v

---

**Referencias**

**86**

## Índice de cuadros

2.10.1. Valores de frecuencias para bandas de Octava en $Hz$ . . . . .	16
2.11.1. Valores de frecuencias para bandas de tercios Octava en $Hz$ . . .	18
2.12.1. Valores de frecuencias para bandas de tercios Octava y sus valores asociados para los Filtros A y Z. . . . .	22
3.0.1. Norma de Regulación de la contaminación acústica en Chile. . .	42



# Índice de figuras

2.3.1. Escala de la presión sonora en $Pa$ versus presión sonora en $dB$ . . . . .	5
2.3.2. Curvas Isofónicas. . . . .	5
2.4.1. Forma de onda del canto de <i>Vanellus Chilensis</i> . . . . .	6
2.4.2. Espectro de frecuencias del <i>Vanellus Chilensis</i> 2.4.1. . . . .	7
2.6.1. Descomposición de Señal de audio. . . . .	8
2.8.1. Ventana Hanning. . . . .	10
2.8.2. 1. Largo de salto, 2. Largo de intersección, 3. Largo de ventana. . . . .	11
2.8.3. Forma de Onda original y forma de onda con la función de ventana Hanning. . . . .	11
2.8.4. Transformada de Fourier en rangos de tiempo mediante el uso de ventanas. . . . .	12
2.9.1. Espectrograma del canto del <i>Vallenus Chilensis</i> . . . . .	13
2.10.1. Gráfica de filtros de bandas de octava. . . . .	17
2.11.1. Gráfica de filtros de bandas de tercios de octava. . . . .	19
2.12.1. Filtro de ponderación A y Z. . . . .	21
2.15.1. Diagrama de funcionamiento del aprendizaje supervisado. . . . .	26
2.16.1. Perceptrón con tres entradas, $x_1, x_2, x_3$ y su salida. . . . .	27
2.16.2. Red neuronal de 3 capas. . . . .	27
2.17.1. Gráfica de función sigmoide. . . . .	30
2.19.1. Gráfica de la función $C$ de dos variables. . . . .	31
2.21.1. Representación matricial de una imagen. . . . .	34
2.21.2. Aplicación de la capa ReLU a una matriz de característica. . . . .	36
2.21.3. Aplicación de la Capa de Agrupación Máxima a la matriz de características. . . . .	37
2.21.4. Esquema de una red neuronal convolucional. En 1 tenemos las capas de entradas, en 2 tenemos las capas ocultas y en 3 la capa de salida. . . . .	37
2.21.5. Esquema de una red neuronal convolucional aplicando Dropout a las capas ocultas. . . . .	38
2.21.6. Esquema de funcionamiento de la capa de agrupación media global. . . . .	38
2.21.7. Esquema de funcionamiento de la capa de aplanamiento. . . . .	39
2.21.8. Esquema de funcionamiento al aplicar la función de activación SoftMax. En 1 tenemos la capa de entrada y en 2 la capa de salida. . . . .	40
3.2.1. Raspberry pi modelo 4 B+. . . . .	44

3.2.2.	Respuesta de frecuencia del modelo BY-M1. . . . .	44
3.2.3.	Micrófono BOYA BY-M1. . . . .	45
3.2.4.	Tarjeta de sonido USB Vention - CDLH0. . . . .	45
3.2.5.	Esquema del hardware. . . . .	46
3.4.1.	Raspberry pi con pantalla táctil LCD. . . . .	47
3.4.2.	Diagrama de funcionamiento del Sonómetro de mano. . . . .	49
3.4.3.	Diagrama de funcionamiento completo del sonómetro de mano. . . . .	49
3.5.1.	Diagrama de funcionamiento de sonómetro continuo. . . . .	52
3.5.2.	Arquitectura de Red Neuronal profunda. . . . .	54
3.5.3.	Diagrama de funcionamiento de sonómetro continuo con Modelo de aprendizaje automático. . . . .	55
3.6.1.	Sonómetro Larson Davis Modelo SoundTrack LXT. . . . .	56
3.6.2.	Diagrama de funcionamiento de homologación. . . . .	57
4.1.1.	Especificaciones del sonómetro continuo. . . . .	58
4.1.2.	Montaje Sonómetro continuo. . . . .	59
4.1.3.	Montaje Sonómetro continuo. . . . .	60
4.1.4.	Medición de niveles para el proceso de homologación. . . . .	61
4.1.5.	Niveles de <i>LASeq</i> de ambos sonómetros. . . . .	61
4.1.6.	Aplicación del parámetro de homologación. . . . .	62
4.1.7.	Promedios minutales de <i>LASeq</i> . . . . .	63
4.1.8.	Frecuencia en diferencias de <i>LASeq</i> . . . . .	63
4.1.9.	Visualización de los niveles promedio de sonido registrados en la aplicación web de SoundAnalytics. . . . .	64
4.1.10.	Funcionalidad de acercamiento detallado de los niveles promedio de sonido. . . . .	65
4.1.11.	Funcionalidad de reproducción de audios registrados. . . . .	65
4.1.12.	Comparación de niveles de sonido en el entorno sonoro 1. . . . .	66
4.1.13.	Comparación de niveles de sonido en el entorno sonoro 2. . . . .	66
4.1.14.	Total de sonómetros registrando datos en la aplicación web de SoundAnalytics. . . . .	67
4.1.15.	Modelo M1. Entrenamiento con 50 épocas. . . . .	68
4.1.16.	Modelo M2. Entrenamiento con 100 épocas. . . . .	68
4.1.17.	Modelo M3. Entrenamiento con 150 épocas. . . . .	68
4.1.18.	Modelo M4. Entrenamiento con 200 épocas. . . . .	69
4.1.19.	Matriz de confusión del modelo M3. . . . .	69
4.1.20.	Ejecución del modelo M3 por segundo. . . . .	70
4.1.21.	Ejecución del modelo M3 por segundo, forma de onda y espectrograma del registro. . . . .	71
4.1.22.	Gráfico de estadísticas diaria de clasificaciones realizadas por el modelo M3. . . . .	71
4.1.23.	Visualización de gráfico de estadísticas diaria de clasificaciones realizadas por el modelo M3 en la aplicación web de SoundAnalytics. . . . .	72
4.1.24.	Otro ejemplo de visualización de gráfico de estadísticas diaria de clasificaciones realizadas por el modelo M3. . . . .	72
4.2.1.	Especificaciones del sonómetro de mano. . . . .	73

---

4.2.2. Data, Pestaña 1 de interfaz gráfica. . . . .	74
4.2.3. Pestaña 2 de interfaz gráfica. . . . .	74
4.2.4. Pestaña 3 de interfaz gráfica. . . . .	75
4.2.5. Sonómetro de mano. . . . .	76
4.2.6. Aplicación de registro de audio. . . . .	76
4.2.7. Aplicación de extracción de datos. . . . .	77
4.2.8. Visualización de base de datos de un registro de audio. . . . .	78
4.2.9. Visualización de bandas de tercios de octavas en la base de datos. . . . .	78
4.2.10. Visualización de archivo XLSX del registro de audio. . . . .	79
4.2.11. Carpetas y archivos de audio almacenados por el modelo. . . . .	79
4.2.12. Valores promedios de rendimiento del sonómetro de mano. . . . .	80

# Capítulo 1

## Introducción

Existen muchos tipos de contaminantes, como también instrumentos y medidas para la regulación de estos, uno de los más comunes con el que interactuamos constantemente, es el ruido. El ruido es un sonido molesto o indeseado, en general lo identificamos como un sonido de gran intensidad. Si bien es relativo para cada persona, existen ciertas normas que establecen y determinan el ruido en un entorno sonoro. Un entorno sonoro lo conforman todos los sonidos característicos de un lugar en un momento particular.

El crecimiento de la población ha creado un aumento en la magnitud de ruido en los entornos sonoros, provocado por el crecimiento del tráfico, industrias y la urbanización. El ruido provoca efectos adversos directos y acumulativos que perjudican y degradan la salud [1], no solo de los humanos sino de la fauna de un entorno natural y urbano [2] [3]. El ruido conduce a la pérdida de audición [3] [4], tinnitus [5] [6], hipersensibilidad al sonido [3], enfermedades cardiovasculares [7] [8], hipertensión [9], trastornos del sueño, estrés, retrasos en el aprendizaje infantil [5], entre otros.

Por estas consecuencias, es de urgencia reconocer y establecer medidas para la gestión del ruido ambiental y así proteger la salud de la población y la fauna, aplicando prevención, descontaminación y mitigación de fuentes de ruidos fijas y móviles. El ruido tiene la particularidad de no dejar residuos, es decir, si la fuente de ruido se detiene, acaba el ruido, por lo cual es necesario tener un registro y un análisis de este. En particular, para el estudio y el análisis del impacto que produce el ruido en los entornos sonoros en Chile, se usan instrumentos profesionales certificados llamados sonómetros, los cuales miden los niveles de

presión sonora y que se ajustan a la norma de regulación de la contaminación acústica en Chile, el Decreto Supremo N° 38/11 del Ministerio del Medio Ambiente [10]. El artículo 11 decreta que las mediciones se efectuarán con un sonómetro que cumpla con las exigencias señaladas para las clases 1 o 2, las cuales se establecen en la norma IEC 61672 [11], donde la clase de un sonómetro describe criterios complejos de precisión, rendimiento y calibración que deben cumplir, siendo la clase 1 la más precisa. Los sonómetros entregan nivel de presión sonora en distintos filtros de ponderación, información detallada del comportamiento del sonido respecto a las distintas frecuencias que lo componen, valores máximos, valores mínimos, valores promedios, etc.

Uno de los grandes inconvenientes en la utilización de sonómetros certificados de clase 1 o 2 es su elevado precio. El ruido al ser un contaminante y una problemática que se debe regularizar con mayor énfasis, el tener más dispositivos como los sonómetros para la regulación, registro y medición de diferentes entornos sonoros es una prioridad, por lo cual el objetivo a realizar en este trabajo es principalmente desarrollar dos tipos de sonómetros de bajo costo y lograr una óptima precisión en sus mediciones, añadiendo características que no tienen los sonómetros convencionales como el acceso remoto, mantención remota, respaldo de datos en la nube, interacción con una aplicación web y una de las características más destacadas es la ejecución de modelos de aprendizaje automático para la clasificación de audios en tiempo real, para así poder automatizar y optimizar la caracterización del entorno sonoro.

Reducir en gran medida el costo de fabricación de un sonómetro nos permite poder optar a desarrollar en mayor cantidad este tipo de instrumentos, pudiendo realizar una mayor cantidad de mediciones en las empresas especializadas, también ser accesibles a estudiantes y docentes para investigaciones académicas.

Se desarrollarán dos tipos de sonómetros, un sonómetro de mano, cuya característica principal es ser un dispositivo de medición transportable, y un sonómetro continuo, que ocupa una ubicación fija y mide sin interrupción un entorno sonoro de interés. El desarrollo de estos sonómetros se hizo en conjunto con SoundAnalytics [12] el cual es un trabajo conjunto de las empresas SocioAnalytics y B & R [13], cuyo objetivo es asesorar a empresas y proyectos de desarrollo en el análisis de los entornos sonoros, evaluar el cumplimiento de normativas y estándares, como la norma de regulación de la contaminación acústica en Chile, el Decreto Supremo N° 38/11 del Ministerio del Medio Ambiente [10].

# Capítulo 2

## Marco Teórico

### 2.1. Onda acústica

La onda acústica u onda de sonido es una onda longitudinal que propaga energía a través de un medio, en forma de onda periódica o cuasiperiódica con carga y descarga adiabática, que transmite lo que se asocia con sonido. En termodinámica, un proceso adiabático se refiere a cuando un proceso termodinámico ocurre sin transferir calor o masa entre el sistema termodinámico y su entorno.

Las ondas acústicas viajan con una velocidad acústica característica, que dependerá siempre del medio por el cual se propaguen. Si se propaga en un medio elástico y continuo, genera **variaciones de presiones** y variaciones de densidades locales. Estas variaciones de densidades, presiones, temperatura y humedad ambiente provocan vibraciones y desplazamientos en las moléculas, donde estas transmiten la vibración a las moléculas vecinas, generando un movimiento en cadena, cuya diferencia de presión generadas por el movimiento y propagación de las moléculas en el medio, producen en el oído humano una interacción y sensación a la cual llamamos sonido [14].

### 2.2. Presión sonora

La presión sonora o acústica, corresponde a la desviación de presión local de la presión atmosférica ambiental causada por una onda sonora, que podemos medir en pascuales ( $Pa$ ), ya que la presión sonora es la fuerza sonora medida en Newton ( $N$ ), que actúa sobre la superficie ( $m^2$ ) perpendicular a la dirección del sonido

[15], es decir:

$$\frac{N}{m^2} = Pa \quad (2.2.1)$$

la presión sonora  $p$  la podemos definir como:

$$p_{total} = p_{local} + p \quad (2.2.2)$$

## 2.3. Nivel de presión sonora

El nivel de presión sonora o acústica, es una medida logarítmica de la presión efectiva de un sonido, respecto a un valor de referencia. La presión sonora más baja posible de escuchar para un humano joven con audición intacta es aproximadamente de  $20\mu Pa$  a una frecuencia de  $1000Hz$  [16], por otro lado, el dolor para el oído humano se experimenta a una presión sonora de  $60Pa$ . Debido al extenso rango entre este umbral, es útil expresar la presión sonora con una escala logarítmica. Este umbral se conoce como el umbral absoluto de audición [17].

Llamemos a  $p_0$  la presión de referencia debido a la presión de un tono apenas audible a  $1000 Hz$ , es decir los  $20\mu Pa$  y  $p$  la presión sonora. Dado que la energía en la onda sonora es proporcional al cuadrado de la presión [16], podemos definir el nivel de presión sonora como:

$$L_p = 10 \log \left( \frac{p^2}{p_0^2} \right) \quad (2.3.1)$$

que es equivalente a:

$$L_p = 20 \log \left( \frac{p}{p_0} \right) \quad (2.3.2)$$

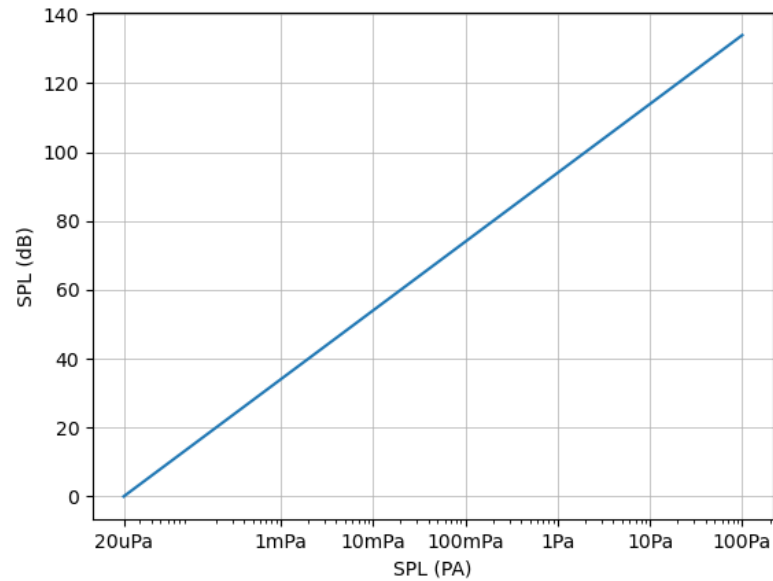
La unidad para expresar el nivel de presión sonora es el decibel ( $dB$ ) [18].

Los niveles de presión sonora audibles en decibeles varían desde los  $0 dB$  y  $120 dB$ , luego de  $120 dB$  en general se producen daños auditivos.

Tenemos entonces que la presión sonora más baja posible de escuchar en decibeles es:

$$20\mu Pa = 0dB \quad (2.3.3)$$

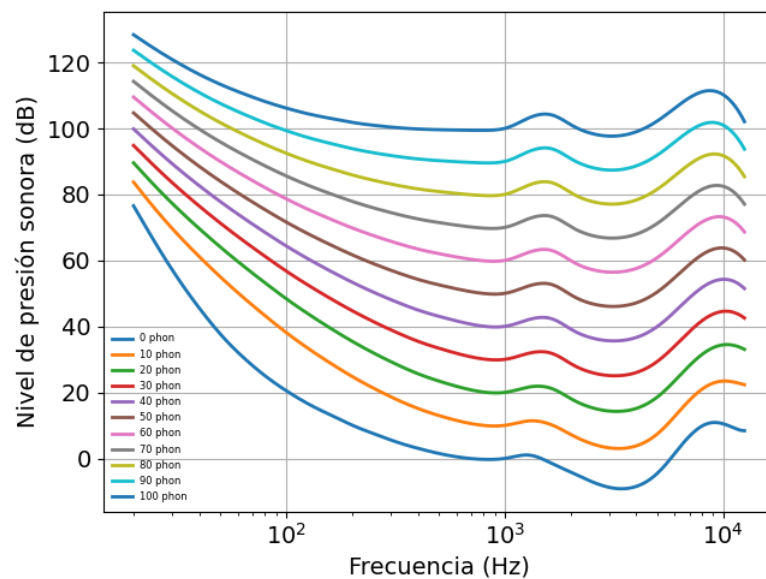
Mediante la ecuación 2.3.1 obtenemos la siguiente gráfica:



**Figura 2.3.1:** Escala de la presión sonora en  $Pa$  versus presión sonora en  $dB$ .

Los oídos humanos si bien detectan cambios en la presión sonora, no tienen la capacidad de percibir de igual manera las distintas frecuencias. Se suele percibir las frecuencias entre  $3000\text{Hz}$  y  $5000\text{Hz}$  de mejor manera que frecuencias bajas y altas.

Podemos observar de la siguiente gráfica:



**Figura 2.3.2:** Curvas Isofónicas.

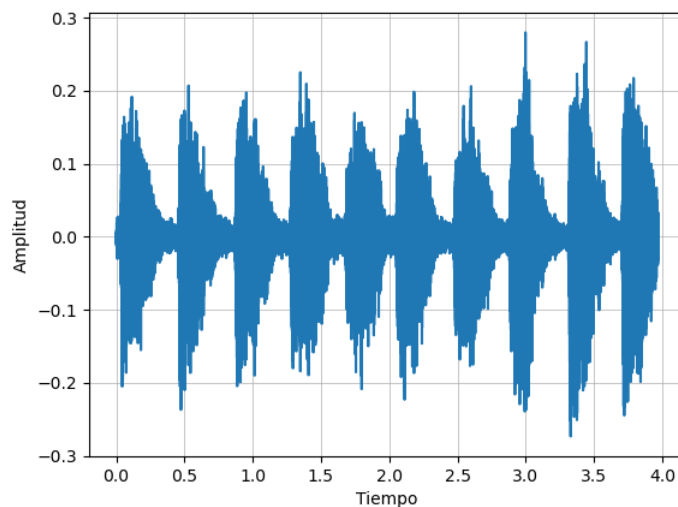


La gráfica de la figura 2.3.2, corresponde a como el oído humano aprecia las diferentes frecuencias del espectro sonoro a diferentes niveles de presión sonora. La unidad de medida para los niveles de sonoridad es el **phon**. Estas curvas están definidas por la norma **ISO 226** [19]. De la primera curva vista en la gráfica, 0 phon, es donde comienza el umbral de escucha del oído humano, y luego de los 100 phon, comienza el dolor en el oído humano. Notamos de esta gráfica que el oído humano, suele escuchar de mejor manera las frecuencias medias, por otro lado, para poder escuchar frecuencias bajas, necesitamos que estén a un rango mucho mayor de decibeles. A  $10dB$  podemos fácilmente escuchar una frecuencia de  $1000Hz$  como se aprecia en la gráfica, pero para poder escuchar una frecuencia de  $30Hz$ , podremos sentirla de forma ideal a los  $60dB$ .

## 2.4. Espectro de frecuencias

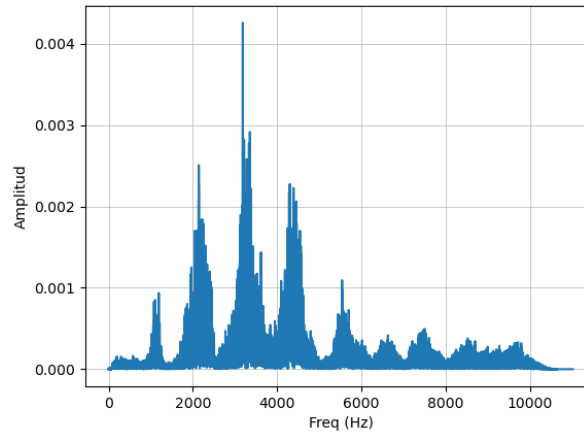
Una onda sonora puede ser la superposición de muchas frecuencias diferentes, la suma de todas estas frecuencias produce un sonido particular, de modo que, al obtener un espectro de frecuencia, podemos realizar un análisis más exhaustivo para ciertas frecuencias de interés.

Podemos observar en la siguiente gráfica una forma de onda del canto del Vanellus Chilensis, ave chilena conocida como Tero o Queltehue [20]:



**Figura 2.4.1:** Forma de onda del canto de Vanellus Chilensis.

Para obtener el espectro de frecuencias y realizar un análisis espectral, significa descomponer la función de onda que representa un sonido en particular, en todas las frecuencias y sus respectivas amplitudes que lo componen.



**Figura 2.4.2:** Espectro de frecuencias del Vanellus Chilensis 2.4.1.

El espectro de frecuencias sonoras nos muestra la distribución de amplitudes para cada una de las frecuencias de un fenómeno ondulatorio, en particular en la gráfica de la figura 2.4.2 vemos para una onda sonora, su descomposición en frecuencias.

Para obtener y efectuar el análisis de frecuencias, usaremos la Transformada de Fourier.

## 2.5. Transformada de Fourier

La Transformada de Fourier, es una transformación matemática que nos permite descomponer a una función en componentes de frecuencias.

La transformada de Fourier de una función se compone de valores complejos a través de sinusoides complejos que componen la función original. Al aplicar una transformada de Fourier a una función, pasamos del dominio del tiempo o espacio, a un dominio de frecuencia temporal.

Para cada frecuencia, el valor absoluto del valor complejo, representará la amplitud de la sinusoide compleja en esa frecuencia, si la frecuencia no existe en el dominio,

la transformación será cero para la frecuencia [15].

Entonces cuando escuchamos un sonido, realmente estamos escuchando ondas sinusoidales de diferentes frecuencias que se superponen. Estas ondas sinusoidales se denominan armónicos del sonido original.

La transformada de Fourier de una función  $f(x)$  se denota por el número complejo:

$$F(w) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t)e^{-iwt} dt \quad (2.5.1)$$

De la gráfica de la figura 2.4.2 podemos apreciar cómo se ven las diferentes frecuencias que componen a la onda de sonido y podemos apreciar los armónicos que componen el canto del Vallenus Chilensis [20].

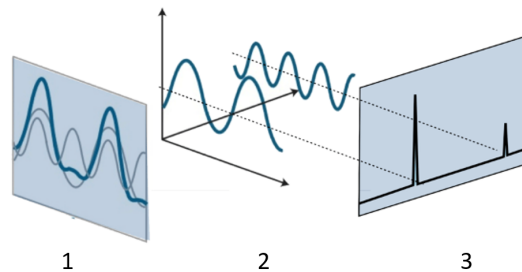
## 2.6. Transformada discreta de Fourier

La transformada discreta de Fourier o DFT (Discrete Fourier Transform) al igual que la Transformada de Fourier 2.5, es una transformación matemática que se aplica a una función para obtener una representación en el dominio de la frecuencia a partir de un dominio del tiempo, pero con la diferencia de que la DFT se aplica solo en funciones que sean discretas y de duración finita [15].

Dada una secuencia de  $N$  números complejos  $a_0, a_1, a_{N-1}$ , mediante la DFT podemos transformar en una secuencia de  $N$  números complejos  $A_0, A_1, A_{N-1}$  a través de:

$$A_k = \sum_{n=0}^{N-1} a_n e^{-\frac{2\pi i}{N} kn} \quad (2.6.1)$$

Con  $k = 0, \dots, N - 1$ . Podemos apreciar en la siguiente gráfica:



**Figura 2.6.1:** Descomposición de Señal de audio.

Tenemos que en la figura 2.6.1, 1. Corresponde a la forma de onda original, en

2 podemos ver la descomposición de la onda original, donde esta se compone de la combinación de dos ondas con frecuencias específicas que se muestra en 3, el plano de las frecuencias.

Para realizar de forma eficiente el cálculo de la DFT, usaremos el algoritmo de la **Transformada rápida de Fourier**.

## 2.7. Transformada rápida de Fourier

La transformada rápida de Fourier o FFT (Fast Fourier Transform) es un algoritmo que nos facilita el cálculo de la DFT con ciertas limitaciones.

Existen muchos algoritmos para el cálculo de las DFT, el que se usará en los cálculos e implementaciones será el algoritmo de Cooley-Tukey [21] [22].

Este algoritmo usa un paradigma de diseño de algoritmos llamado divide y vencerás, que corresponde a dividir recursivamente un problema en dos o más subproblemas relacionados, hasta que estos se vuelven lo suficientemente simples para que sean resueltos de forma directa, para luego combinar las soluciones y dar una solución al problema general.

El cálculo de la DFT se descompone en otras DFT más sencillas hasta poder llegar a transformadas de 2 elementos donde  $k$  en la ecuación 2.6.1 puede tomar los valores de 0 y 1, y luego agrupar estos resultados hacia los niveles superiores y así resolver sucesivamente hasta llegar al resultado final.

## 2.8. Transformada de Fourier de tiempo reducido

La transformada de Fourier de tiempo corto o de tiempo reducido (STFT: Short Time Fourier Transform) es un método de descomposición espectral para una señal variable, que implica calcular la transformada de Fourier a través del tiempo, mediante cortas ventanas de tiempo, las cuales desplazamos con intervalos determinados.

Cuando calculamos la Transformada de Fourier 2.5, como por ejemplo en 2.4.2, tenemos un solo espectro de frecuencias para los 4 segundos que dura aproximadamente esa muestra, de modo que, no tenemos la información de como van variando las frecuencias en cortos intervalos de tiempo, debido a que en el principio de la señal puede que interactúen diferentes frecuencias a las que interactúan al final, pero con un análisis espectral general, no tenemos información

de aquello. Por otro lado, la STFT nos permite a través de pequeñas ventanas de tiempo, obtener la información de que frecuencias están presentes en ese instante y nos muestra un esquema más específico y no general, de cómo actúa la Transformada de Fourier, por lo cual cuando calculamos la STFT, estamos calculando una secuencia de Transformadas de Fourier de una señal a través del tiempo [15].

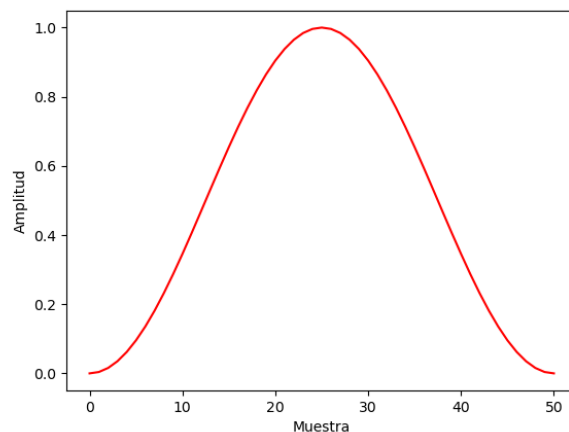
Para casos de tiempo discreto, los datos que se transforman se pueden dividir en bloques que pueden superponerse entre si, donde en cada bloque se aplicará la transformada de Fourier y cada resultado complejo se agrega a una matriz, la cual tiene la magnitud y fase de cada punto en el tiempo y su frecuencia asociada. Para el cálculo de las transformadas de Fourier, nuevamente hacemos uso del algoritmo FFT 2.7.

La señal original es multiplicada por una función de ventana, para poder recorrer la función original y evitar discontinuidades al principio y final de los bloques en que calcularemos la transformada de Fourier. Esta función de ventana se desplazará a través del tiempo y se calculará la transformada de Fourier para cada una de las señales ventana.

Una de las funciones ventana más utilizada es la ventana Hanning [20], la cual se define como:

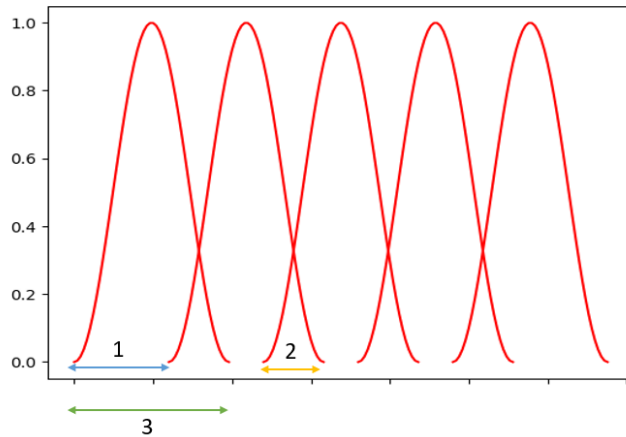
$$v(n) = a_0 - a_1 \cos \frac{2\pi n}{N-1} \quad (2.8.1)$$

con  $a_0 = 0,5$ ,  $a_1 = 0,5$  y  $0 \leq n \leq M-1$ . La gráfica de esta función es representada por:



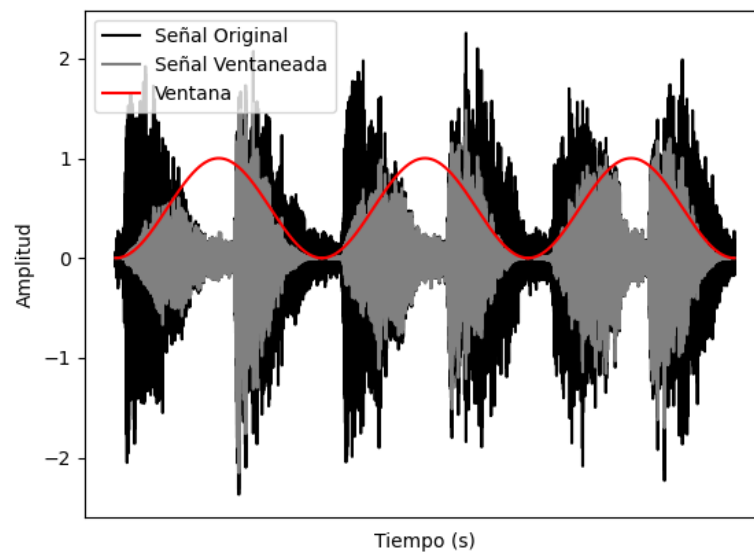
**Figura 2.8.1:** Ventana Hanning.

Donde esta puede ser superpuesta a través del tiempo y tener parámetros importantes para la aplicación correcta:



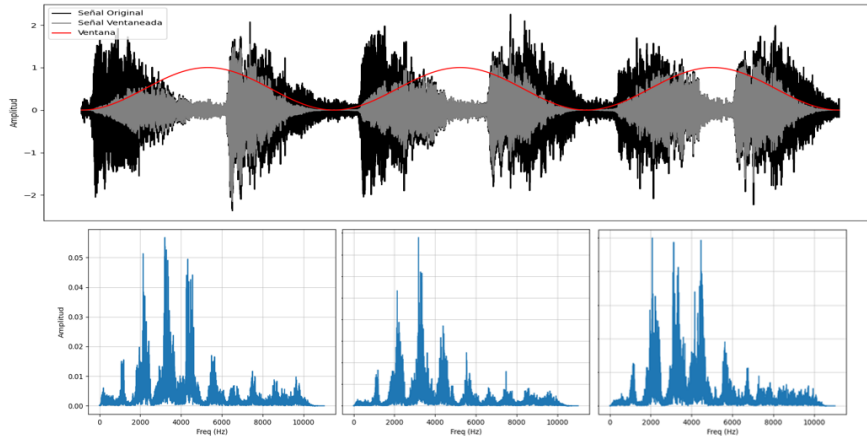
**Figura 2.8.2:** 1. Largo de salto, 2. Largo de intersección, 3. Largo de ventana.

donde 1 representa la distancia entre el inicio de dos ventanas consecutivas, 2 la distancia de la intersección que ocurre entre ventanas y 3 el largo de la ventana. Al aplicar estas ventanas, por ejemplo, a la forma de onda 2.4.1, que representa el canto del Vallenus Chilensis [20], obtenemos:



**Figura 2.8.3:** Forma de Onda original y forma de onda con la función de ventana Hanning.

Podemos observar cómo se transforma la forma de onda a través del uso de la función ventana de Hanning. Para cada una de las ventanas, se aplicará la Transformada de Fourier mediante el algoritmo FFT:



**Figura 2.8.4:** Transformada de Fourier en rangos de tiempo mediante el uso de ventanas.

observamos que para cada bloque de datos donde se aplicó la función ventana, podemos obtener un espectro de frecuencia y podemos ser capaces de ver cuáles son las frecuencias que interactúan en determinados tiempos. Si bien en este ejemplo no se usaron pequeños rangos de tiempo y el ejemplo gráfico no cuenta con un largo de interacción, para tiempos muy cortos la aplicación es la misma y es aún más detallado la variación de frecuencia durante el tiempo. Podemos ver entonces que la STFT dependerá siempre de las propiedades de la señal y de la función ventana que se elija para obtener la transformada de Fourier.

## 2.9. Espectrograma

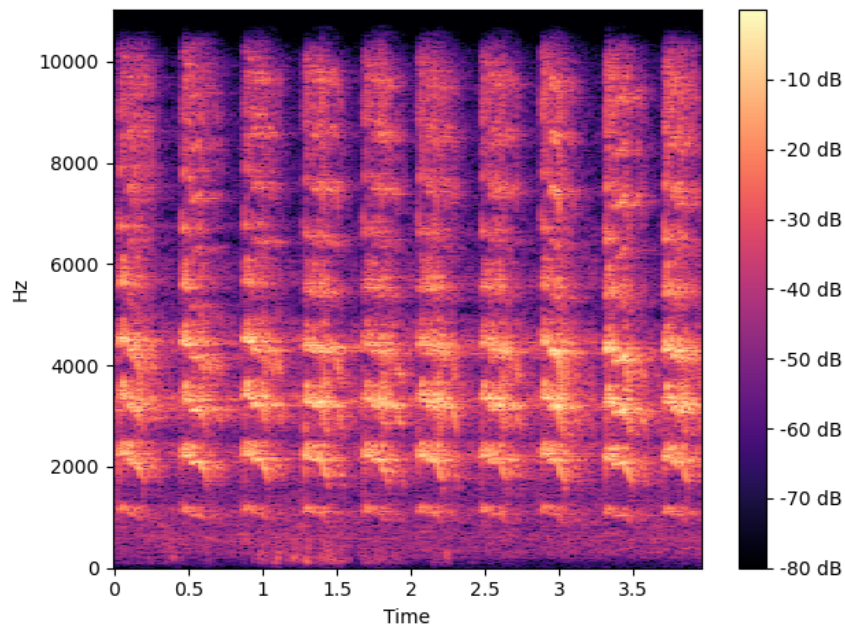
El espectrograma es una representación visual del espectro de frecuencias de una señal a través del tiempo. Podemos crear un espectrograma mediante un proceso digital, mediante la STFT 2.8, para luego poder calcular la magnitud del espectro de frecuencias para cada bloque, donde cada uno de esos bloques corresponderá a una línea vertical de las que componen el espectrograma, es decir cada línea vertical del espectrograma representa una medida de magnitud versus frecuencia para un tiempo específico, al juntar cada uno de estos bloques podemos formar la imagen que representa al comportamiento de las frecuencias y su magnitud a

través del tiempo [15].

Para calcular la matriz del espectrograma, debemos obtener la magnitud de los valores de la STFT, es decir:

$$\text{espectrograma} = |STFT(t)|^2 \quad (2.9.1)$$

Para una señal dada, con la ecuación 2.9.1 obtenemos su espectrograma, el cual nos muestra el dominio frecuencial de la señal, de la variación de la energía y la frecuencia en función del tiempo. Para la forma de onda 2.4.1, que representa el canto del Vallenus Chilensis, su espectrograma obtenido a través de la ecuación 2.9.1, nos queda:



**Figura 2.9.1:** Espectrograma del canto del Vallenus Chilensis.

Esta gráfica de tres dimensiones para este caso es presentada como una imagen de dos dimensiones, donde en el eje de abscisas tenemos el tiempo en el eje de ordenadas las frecuencias y a representación de la energía en  $dB$  en el plano tridimensional, donde tenemos un rango de colores que indicarán los niveles de la energía. Entonces podemos observar el cambio de frecuencias a través del tiempo, para ventanas muy cortas de tiempo, donde es calculada la transformada de Fourier. Podemos apreciar del canto, las frecuencias que lo componen y sus armónicos. Esta gráfica corresponde a 4 segundos de canto continuo y con un



rango de  $0Hz$  a  $10,000Hz$ .

## 2.10. Bandas de Octava

Para poder analizar de una manera más precisa el gran espectro de frecuencia que puede tener una señal, se hace uso de las bandas. Las bandas son un método de filtrado para dividir el espectro en segmentos más pequeños. En particular para el espectro de sonido, se usan las bandas de octavas.

Dividir un espectro en bandas de octava nos ayudará a obtener información más detallada por cada una de esas bandas para las frecuencias de interés. Los parámetros para obtener las bandas y otras definiciones necesarias, están dadas por la norma internacional ANSI S1.11-2004 (ANSI: American National Standards Institute ) [23].

### 2.10.1. Filtro paso banda

Un filtro paso banda, es un filtro electrónico que permite pasar un determinado rango de frecuencias de señal, que se extiende desde una frecuencia de banda inferior, pasando por una frecuencia de banda central hasta una frecuencia de banda superior finita.

Para poder calcular la banda de octava, debemos definir ciertos parámetros.

### 2.10.2. Relación de Octavas

Se define la relación de octavas u Octave ratio para un sistema de base 10 como:

$$G_{10} = 10^{3/10} \quad (2.10.1)$$

Y para base dos como:

$$G_2 = 2 \quad (2.10.2)$$

Ecuaciones que nos sirven para determinar una relación de frecuencias de banda de octava o banda de octava fraccionaria.

### 2.10.3. Designador de ancho de banda

Se define como el recíproco de un entero positivo, para designar la fracción de una banda de octava, denotado como  $\frac{1}{b}$

### 2.10.4. Frecuencia de referencia

La Frecuencia de referencia es  $1000Hz$ , la cual denotamos como  $f_r$

### 2.10.5. Frecuencia media exacta

La Frecuencia media exacta de banda, tiene relación específica con la frecuencia de referencia 2.10.4 y del término  $G$  de la relación de octavas 2.10.2. La relación de las frecuencias exactas de banda media de dos filtros de paso de banda contiguos cualesquiera, de una anchura de banda de octava fraccionaria especificada, es la misma.

Cuando el denominador del designado de ancho de banda 2.10.3 es un número impar, la frecuencia media de banda de cualquier filtro, se determina mediante:

$$f_m = (G^{(x-30)/b})(f_r) \quad (2.10.3)$$

y cuando el denominador del designado de ancho de banda 2.10.3 es un número par:

$$f_m = (G^{(2x-29)/b})(f_r) \quad (2.10.4)$$

#### 2.10.5.1. Frecuencia media nominal

Es la frecuencia media exacta redondeada en su valor.

#### 2.10.5.2. Frecuencias del borde

Existen dos tipos de frecuencia de borde, la inferior y superior,  $f_1$  será la frecuencia inferior y  $f_2$  a la frecuencia superior, y las definimos como:

$$f_1 = (G^{-1/(2b)})(f_m) \quad (2.10.5)$$

$$f_2 = (G^{+1/(2b)})(f_m) \quad (2.10.6)$$

con 2.10.5 y 2.10.6 podemos definir a  $f_m = \sqrt{f_1 f_2}$

### 2.10.5.3. Ancho de banda

Definimos mediante 2.10.5 y 2.10.6 el ancho de banda como  $f_2 - f_1$

### 2.10.6. Banda de Octava

Una banda de octava se define como una banda donde la frecuencia más alta, equivale a dos veces la frecuencia más baja.

### 2.10.7. Bandas fraccionarias de octava

Una banda fraccionaria de octava se define para la relación entre la frecuencia de borde superior 2.10.6 y la frecuencia de borde inferior 2.10.5 es una octava elevada a un exponente correspondiente al designador de ancho de banda 2.10.3, tal que  $f_1/f_2 = G^{1/b}$ .

Con las ecuaciones anteriormente definidas en citar el tema banda, podemos obtener las bandas de octavas, donde:

$$G = 10^{3/10} \quad (2.10.7)$$

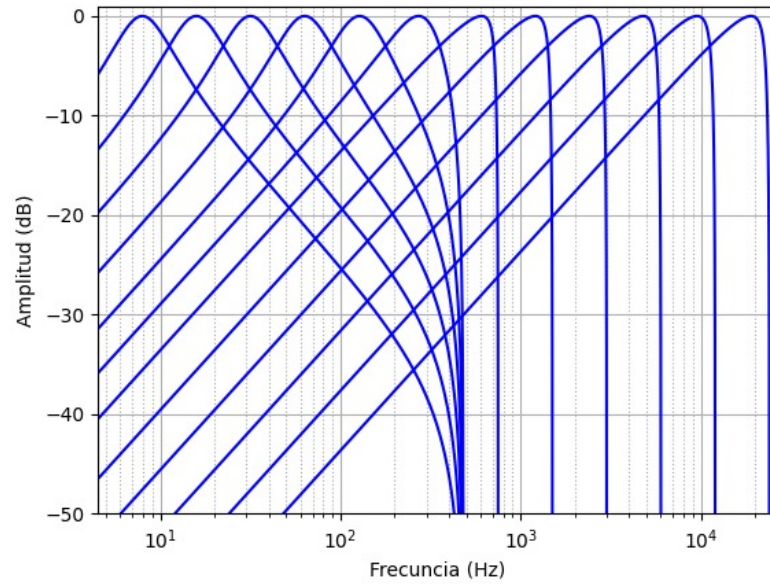
$$f_m = (G^{(x-30)/1})(1000Hz) \quad (2.10.8)$$

$$f_1 = (G^{-1/2})(1000Hz) \quad (2.10.9)$$

Y obtenemos las siguientes bandas de octava:

Bandas de Octava				
Banda	$f_1$	$f_m$	$f_2$	$f_m$ nominal
1	5.62	7.94	11.22	8
2	11.22	15.85	22.39	16
3	22.39	31.62	44.67	31.5
4	44.67	63.1	89.13	63
5	89.13	125.89	177.83	125
6	177.83	251.19	354.81	250
7	354.81	501.19	707.95	500
8	707.95	1000.0	1412.54	1000
9	1412.54	1995.26	2818.38	2000
10	2818.38	3981.07	5623.41	4000
11	5623.41	7943.28	11220.18	8000
12	11220.18	15848.93	22387.21	16000

**Cuadro 2.10.1:** Valores de frecuencias para bandas de Octava en  $Hz$ .



**Figura 2.10.1:** Gráfica de filtros de bandas de octava.

## 2.11. Bandas de tercios de octava

Para el cálculo de las bandas de tercios de octava, tenemos:

$$G = 10^{3/10} \quad (2.11.1)$$

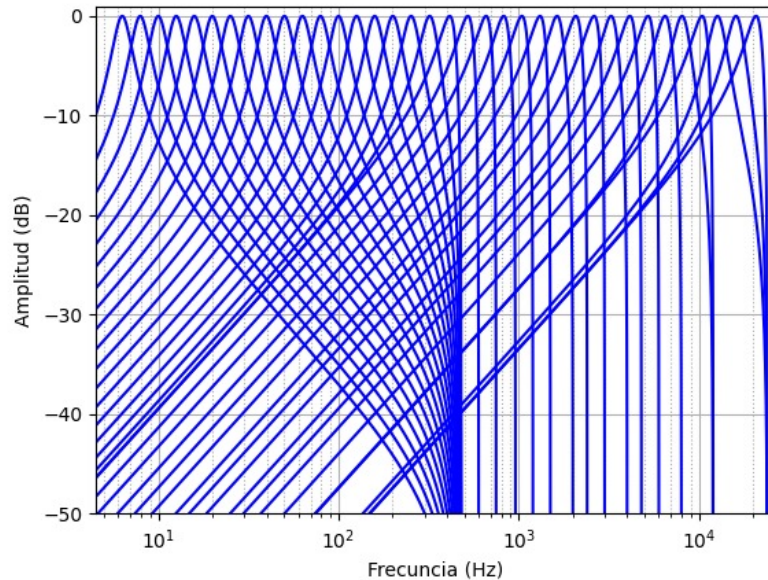
$$f_m = (G^{(x-30)/3})(1000Hz) \quad (2.11.2)$$

$$f_1 = (G^{-1/6})(1000Hz) \quad (2.11.3)$$

Y obtenemos las siguientes bandas de tercios de octava:

<b>Bandas de Tercios de Octava</b>				
<b>Banda</b>	$f_1$	$f_m$	$f_2$	$f_m$ <b>nominal</b>
1	5.62	6.31	7.08	6.3
2	7.08	7.94	8.91	8
3	8.91	10.0	11.22	10
4	11.22	12.59	14.13	12.5
5	14.13	15.85	17.78	16
6	17.78	19.95	22.39	20
7	22.39	25.12	28.18	25
8	28.18	31.62	35.48	31.5
9	35.48	39.81	44.67	40
10	44.67	50.12	56.23	50
11	56.23	63.1	70.79	63
12	70.79	79.43	89.13	80
13	89.13	100.0	112.2	100
14	112.2	125.89	141.25	125
15	141.25	158.49	177.83	160
16	177.83	199.53	223.87	200
17	223.87	251.19	281.84	250
18	281.84	316.23	354.81	315
19	354.81	398.11	446.68	400
20	446.68	501.19	562.34	500
21	562.34	630.96	707.95	630
22	707.95	794.33	891.25	800
23	891.25	1000.0	1122.02	1000
24	1122.02	1258.93	1412.54	1250
25	1412.54	1584.89	1778.28	1600
26	1778.28	1995.26	2238.72	2000
27	2238.72	2511.89	2818.38	2500
28	2818.38	3162.28	3548.13	3150
29	3548.13	3981.07	4466.84	4000
30	4466.84	5011.87	5623.41	5000
31	5623.41	6309.57	7079.46	6300
32	7079.46	7943.28	8912.51	8000
33	8912.51	10000.0	11220.18	10000
34	11220.18	12589.25	14125.38	12500
35	14125.38	15848.93	17782.79	16000
36	17782.79	19952.62	22387.21	20000

**Cuadro 2.11.1:** Valores de frecuencias para bandas de tercios Octava en  $Hz$ .



**Figura 2.11.1:** Gráfica de filtros de bandas de tercios de octava.

## 2.12. Filtros de ponderación

Como se mencionó en 2.3, los oídos humanos si bien detectan cambios en la presión sonora, no tienen la capacidad de percibir de igual manera las distintas frecuencias. Los filtros de ponderación son utilizados para resaltar características respecto a otras, en particular para una señal de audio, nos permite resaltar ciertas frecuencias y ocultar otras, en consecuencia nos permite una medición ponderada en lugar de una medición plana respecto al nivel de energía.

Cuando se realizan mediciones y análisis sonoros mediante instrumentos de medición, como el sonómetro, se aplican ciertos filtros de ponderación para poder capturar el sonido y el nivel de sonido de manera similar a la percepción humana, para así percibir lo que probablemente escucharía un oído humano con una audición normal, y poder ejecutar un análisis en base a nuestra manera de escuchar.

Definiremos dos filtros de ponderación de frecuencia diferentes descritos en ISO 226:2003 [19] y [11]

### 2.12.1. Filtro de ponderación Z

El filtro de ponderación Z corresponde a una respuesta plana respecto a las frecuencias, en cero decibeles, lo que equivale a no tener ningún tipo de filtro.

### 2.12.2. Filtro de ponderación A

El filtro de ponderación A es utilizado en mediciones de ruido ambiental y es el tipo de medición estándar en instrumentos de mediciones sonoras. Se define en la norma internacional IEC 61672:2003 [11] y es usado por la mayoría de las normas nacionales e internacionales relacionadas con la medición de nivel de presión sonora [10].

El nivel de sonido medido con filtro de ponderación A representa la sensibilidad del oído humano en función de la frecuencia de los tonos puros. El filtro de ponderación A se basa en las curvas isofónicas 2.3.2 de 40 phon [19] [24].

Se suele definir en una tabla de valores, el filtro para bandas de octavas y tercios de octava, donde a los valores medidos de presión sonora se agrega el valor del filtro asociado a cada banda.

Definimos el filtro de ponderación A como:

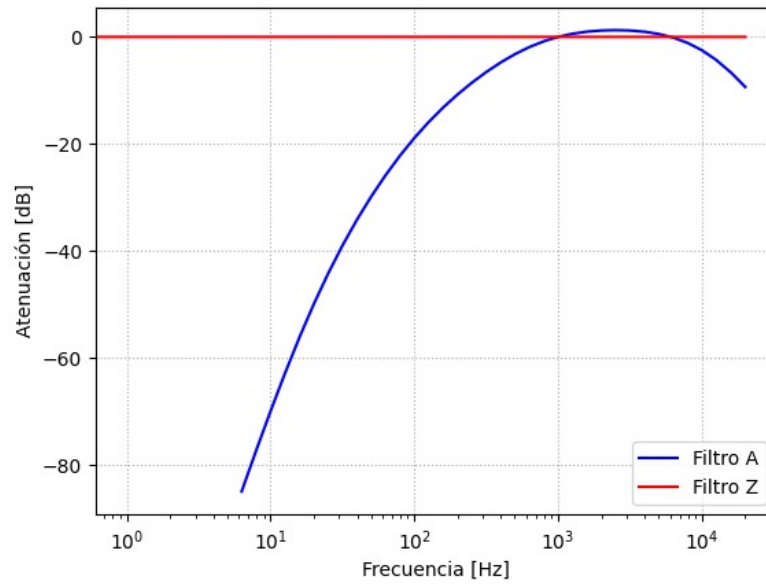
$$A(f) = 20 \log R_a(f) - 20 \log R_a(1000) \quad (2.12.1)$$

$$A(f) \approx 20 \log R_a(f) + 2 \quad (2.12.2)$$

donde  $R_a(f)$  es:

$$R_a f = \frac{12194^2 f^4}{(f^2 + 20,6^2) \sqrt{(f^2 + 107,7^2)(f^2 + 737,9^2)(f^2 + 12194^2)}} \quad (2.12.3)$$

Podemos ver gráficamente el comportamiento de estos filtros y sobre que frecuencias actúan:



**Figura 2.12.1:** Filtro de ponderación A y Z.



Con los filtros definidos, para cada banda tenemos que:

Bandas de Tercios de Octava						
Banda	$f_1$	$f_m$	$f_2$	$f_m$ nominal	Filtro Z	Filtro A
1	5.62	6.31	7.08	6.3	0.0	-85.4
2	7.08	7.94	8.91	8	0.0	-77.8
3	8.91	10.0	11.22	10	0.0	-70.4
4	11.22	12.59	14.13	12.5	0.0	-63.4
5	14.13	15.85	17.78	16	0.0	-56.7
6	17.78	19.95	22.39	20	0.0	50.5
7	22.39	25.12	28.18	25	0.0	-44.7
8	28.18	31.62	35.48	31.5	0.0	-39.4
9	35.48	39.81	44.67	40	0.0	-34.6
10	44.67	50.12	56.23	50	0.0	-30.2
11	56.23	63.1	70.79	63	0.0	-26.2
12	70.79	79.43	89.13	80	0.0	-22.5
13	89.13	100.0	112.2	100	0.0	-19.1
14	112.2	125.89	141.25	125	0.0	-16.1
15	141.25	158.49	177.83	160	0.0	-13.4
16	177.83	199.53	223.87	200	0.0	-10.9
17	223.87	251.19	281.84	250	0.0	-8.6
18	281.84	316.23	354.81	315	0.0	-6.6
19	354.81	398.11	446.68	400	0.0	-4.8
20	446.68	501.19	562.34	500	0.0	-3.2
21	562.34	630.96	707.95	630	0.0	-1.9
22	707.95	794.33	891.25	800	0.0	-0.8
23	891.25	1000.0	1122.02	1000	0.0	0
24	1122.02	1258.93	1412.54	1250	0.0	0.6
25	1412.54	1584.89	1778.28	1600	0.0	1.0
26	1778.28	1995.26	2238.72	2000	0.0	1.2
27	2238.72	2511.89	2818.38	2500	0.0	1.3
28	2818.38	3162.28	3548.13	3150	0.0	1.2
29	3548.13	3981.07	4466.84	4000	0.0	1.0
30	4466.84	5011.87	5623.41	5000	0.0	0.5
31	5623.41	6309.57	7079.46	6300	0.0	-0.1
32	7079.46	7943.28	8912.51	8000	0.0	-1.1
33	8912.51	10000.0	11220.18	10000	0.0	-2.5
34	11220.18	12589.25	14125.38	12500	0.0	-4.3
35	14125.38	15848.93	17782.79	16000	0.0	-6.6
36	17782.79	19952.62	22387.21	20000	0.0	-9.3

**Cuadro 2.12.1:** Valores de frecuencias para bandas de tercios Octava y sus valores asociados para los Filtros A y Z.

## 2.13. Ponderación de tiempo

El sonido se mide detectando los cambios en la presión sonora, donde estos niveles cambian con rapidez en la mayoría de entornos sonoros. La ponderación de tiempo amortigua los cambios repentinos en los niveles, creando así una visualización más suave. Existen varias atenuaciones posibles definidas en la norma IEC 61672-1 [11], en particular se usará la suavizado exponencial lenta, que se denota con la letra **S** de Slow. Al medir promedios, S es lo más apropiado cuando se miden señales que oscilan rápidamente, lo que es común en entornos sonoros.

Sea  $x_t$  una secuencia de datos, comenzando en un tiempo  $t = 0$ . El suavizado exponencial se define como:

$$S_t = \alpha x_t + (1 - \alpha) s_{t-1} \quad (2.13.1)$$

donde  $\alpha$  es el factor de suavizado.

## 2.14. Nivel de presión sonora continuo equivalente

El nivel de sonido equivalente ( $L_{eq}$ ) cuantifica el sonido a un solo valor de nivel de sonido para cualquier duración de tiempo deseada. Es utilizado para representar el nivel de sonido de una fuente variable durante un tiempo determinado. Se conoce también como sonido promedio.

Calculamos el nivel de Sonido equivalente como:

$$L_{eq} = 10 \log \left( \frac{1}{N} \sum_{i=1}^N 10^{\left(\frac{L_{eq_i}}{10}\right)} \right) \quad (2.14.1)$$

Con  $N$  el número de muestras en el intervalo de tiempo de referencia y  $L_{eq,T}$  los niveles de presión sonora en el tiempo.

Al calcular el nivel de presión sonora equivalente en  $dB$  usando filtros de ponderación y ponderación temporal  $S$ , se denotará al nivel sonoro equivalente con filtro de ponderación A como  $L_{ASeq}$  y  $L_{ZSeq}$  al nivel sonoro equivalente con ponderación Z.

## 2.15. Aprendizaje automático

El aprendizaje automático es parte de las ciencias de la computación, de la rama de inteligencia artificial. El objetivo principal, es desarrollar y programar dispositivos capaces de "**aprender**.<sup>a</sup> a través de los datos, mejorar el rendimiento y lograr predicciones precisas, sin ser programados explícitamente. Al igual que con cualquier ejercicio computacional, la calidad y la cantidad de los datos serán cruciales para la precisión de las predicciones que se realizarán. Se define el aprender como *“obtener conocimiento, comprensión o habilidad mediante el estudio, la instrucción o la experiencia”* y *“modificación de una tendencia conductual mediante la experiencia”* [25].

Este algoritmo de aprendizaje se basa en la repetición. Recuerde que un algoritmo no es más que un conjunto de instrucciones que una computadora utiliza para transformar una entrada, en una salida particular. El aspecto de aprendizaje es solo un algoritmo que repite su operación de ejecución una y otra vez, y realiza ligeros ajustes, hasta que se cumple un cierto conjunto de condiciones. La prueba crucial de un algoritmo de aprendizaje es cuando este es capaz de predecir cuándo se le dan nuevos datos, no usados en el entrenamiento. [26]

Entonces, mediante algoritmos y estadísticas, un dispositivo computacional reconoce, cuando puede mejorar a través del uso de los datos, proceso el cual llamamos entrenamiento. Existen muchos caminos para lograr que una computadora aprenda a través de los datos, técnicas que dependerán del objetivo a lograr. En particular, en el aprendizaje automático los datos determinarán en su mayoría de veces, cuáles son las técnicas que se pueden adaptar mejor al uso correcto de ellos. Podemos denotar tres ramas principales del aprendizaje automático:

- Aprendizaje supervisado.
- Aprendizaje no supervisado.
- Aprendizaje por refuerzo.

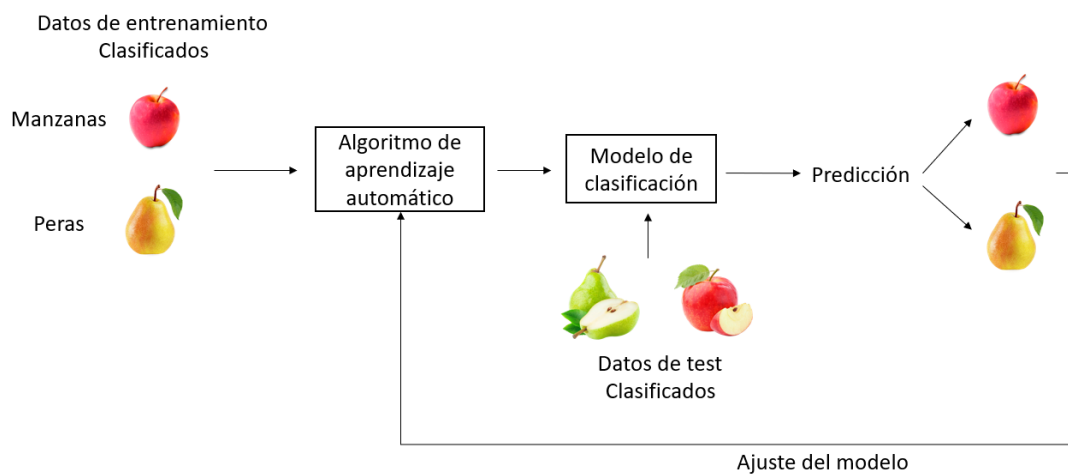
El aprendizaje no supervisado, consiste en la entrega de datos no clasificados. El proceso de aprendizaje en este caso dependerá de los patrones que se analizan repetidamente en los datos, usando métodos de estimación basados en estadísticas inferenciales, para describir patrones, relaciones y correlación con el conjunto de

datos sin procesar. El aprendizaje por refuerzo, al igual que en el Aprendizaje no supervisado, usará datos no clasificados, pero en este caso los patrones detectados se clasifican de manera positiva o negativa, así proporcionando un lazo de retroalimentación al algoritmo, que permite afirmar si las clasificaciones están siendo correctas o no, por lo cual, para este método existe un cierto nivel de supervisión. [26] [27]

### 2.15.1. Aprendizaje Supervisado

Una de las implementaciones en el sonómetro creado para esta tesis, utiliza un modelo de aprendizaje automático, creado por el aprendizaje supervisado.

A diferencia de los otros dos métodos mencionados anteriormente, El aprendizaje supervisado funciona con datos previamente clasificados, antes de ejecutar el entrenamiento. Al entrenar una computadora con distintas clasificaciones de datos, esta puede aprender a través del entrenamiento, a identificar patrones en los datos y realizar predicciones. Los datos clasificados tienen una entrada y salida identificada y esperada. El modelo constantemente se va evaluando respecto a los patrones que detecta, con nuevos datos que están fuera de los datos de entrenamiento, pero que de igual manera están clasificados para revisar el resultado, y reajustar el modelo dependiendo que tan certeros sean los resultados obtenidos. Estos datos suelen llamarse datos de prueba. [27] [28]. Por ejemplo, tenemos datos clasificados tales como, manzanas y peras. Entrenamos un modelo para que aprenda las características de estos dos elementos y comprobamos con nuevos datos si el modelo aprendió los patrones. De este modelo, obtenemos un diagrama de la siguiente forma:



**Figura 2.15.1:** Diagrama de funcionamiento del aprendizaje supervisado.

En la figura 2.15.1 podemos ver el funcionamiento de un modelo supervisado. Existen datos clasificados de entrada para el modelo, luego se extraen sus características, en el algoritmo de aprendizaje automático mediante el entrenamiento, y resulta un modelo de clasificación. El modelo obtenido se toma a prueba con nuevos datos y se evalúa la respuesta del modelo, para reajustarlo y así poder mejorar la precisión. La respuesta será una clasificación, es decir, si detecto pera o manzana como dato de prueba. Este proceso se repite hasta lograr una buena respuesta de clasificación.

Existen muchas técnicas de clasificación para este tipo de modelos, árboles de decisiones, reglas de asociación, máquina de vectores de soporte y por último, la clasificación que da base al desarrollo del modelo utilizado en este trabajo, el algoritmos de redes neuronales artificiales. [27]

### 2.15.2. Redes Neuronales Artificiales

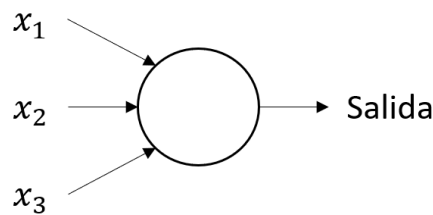
Las redes neuronales artificiales (RNA), inspiradas en el funcionamiento de las redes neuronales biológicas, constan de sistema de enlaces de neuronas, que interactúan entre sí para obtener un estímulo de salida (proceso que ejecuta el modelo para hacer una predicción). Estas conexiones tienen pesos numéricos que se ajustan según la experiencia obtenida del entrenamiento, de esa manera las redes neuronales se adaptan a un impulso (entrada de datos) y son capaces de aprender.

Las redes neuronales suelen estar divididas en distintas capas conectadas entre sí,

que tienen funciones particulares cada una de ellas, donde siempre existen dos capas básicas, la capa de entrada y la capa de salida. [28]

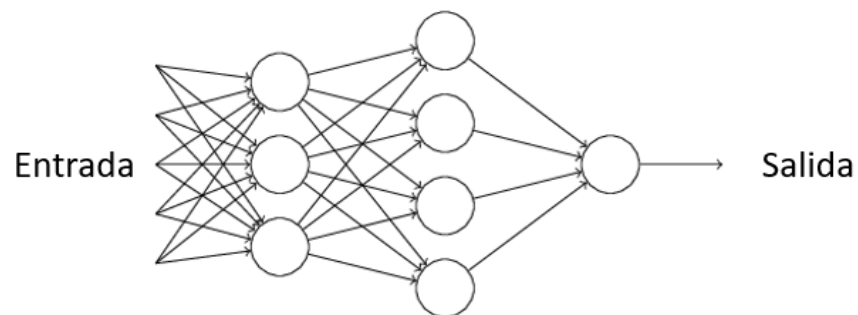
## 2.16. Perceptrón

El perceptrón es un tipo de neurona artificial que toma varias entradas binarias, y produce una única salida binaria [26]:



**Figura 2.16.1:** Perceptrón con tres entradas,  $x_1, x_2, x_3$  y su salida.

Para calcular la salida del perceptrón se ocupan pesos, números reales que expresan la importancia de las respectivas entradas y salidas. Este simple modelo matemático es la base del funcionamiento de las redes neuronales, donde el perceptrón es un elemento que toma decisiones sopesando evidencia. Al existir variaciones en los pesos, el modelo podrá tomar distintas decisiones. Un perceptrón de por sí puede solo tomar decisiones muy simples, en base a la entrada y sus pesos, pero teniendo una red de estas neuronas se puede llegar a tomar decisiones mucho más sutiles.



**Figura 2.16.2:** Red neuronal de 3 capas.

Podemos apreciar de la figura 2.17.1, que esta red neuronal consta de una primera

capa de 3 perceptrones, que están conectados con una capa central, donde cada perceptrón de esta capa recibe información de las capas anteriores, teniendo una sola salida, donde cada uno de estos perceptrones está tomando una decisión más compleja, sopesando los resultados de la primera capa. El perceptrón de la última capa, el cual obtiene toda la información de la red, tiene solo una salida, que dependerá del objetivo del modelo.

Definimos al perceptrón de la siguiente forma:

$$\text{salida} = \begin{cases} 0 & \text{si } w \cdot x + b \leq 0 \\ 1 & \text{si } w \cdot x + b > 0 \end{cases} \quad (2.16.1)$$

Donde la expresión  $\sum_j w_j x_j$  es el producto punto entre los datos de entrada  $x$  y los pesos  $w$ , y  $b = -\text{sesgo}$  del perceptron.

Para un nodo de la red, las entradas se multiplican por los pesos y se suman con el ajuste que se va realizando en la red para mejorar la clasificación, llamado sesgo y denotado como  $b$ . El perceptrón toma valores de uno si el producto punto es positivo y cero si es igual o menos que cero.

Una función que se aplica para la salida de un nodo de la red, la llamaremos función de activación.

## 2.17. Neurona Sigmoide

Según la definición del perceptrón y su salida, sabemos que los valores pueden ser 1 y 0, lo que es un rango un poco radical para cuando queremos que una red pueda aprender con mucha precisión. Un pequeño cambio en un peso o sesgo de la red produce un cambio en la salida de la red, donde un solo perceptrón que varíe, puede provocar de pasar de una salida 1 a 0, lo cual dificulta bajo esta estructura, el modificar los pesos y sesgos para lograr el comportamiento adecuado de la red. Para este problema se introduce la Neurona sigmoide, que es similar al perceptron, pero modificadas para que pequeños cambios en una neurona, tanto como el sesgo y el peso, provoque pequeños cambios en la salida del modelo, lo que permitirá entrenar con mayor precisión una red de neuronas sigmoides.

A diferencia del perceptron, la Neurona Sigmoide puede tomar valores entre 0 y 1,

y esta salida se define como:

$$\sigma(w \cdot x + b) = \frac{1}{1 + \exp - \sum_j w_j x_j - b} \quad (2.17.1)$$

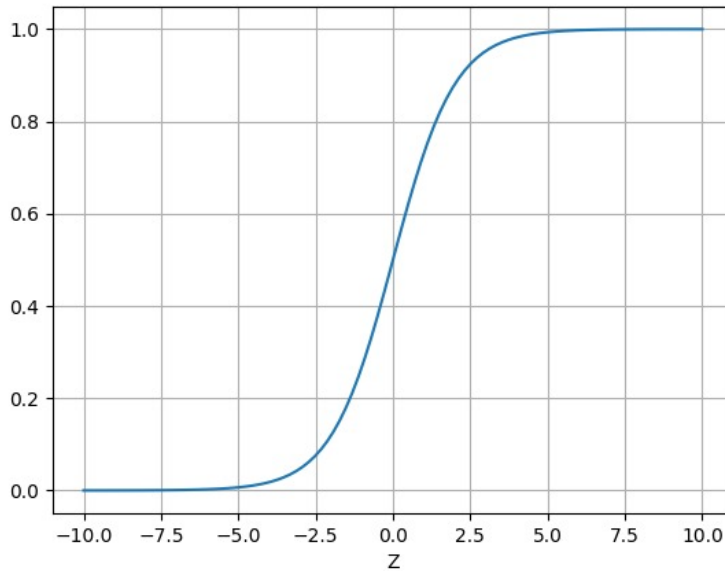
Donde podemos denotar como  $z = - \sum_j w_j x_j - b$  y nos queda:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.17.2)$$

Donde  $\sigma(z)$  es llamada la función sigmoide.



La función sigmoide tiene la siguiente forma:



**Figura 2.17.1:** Gráfica de función sigmoide.

Tenemos que el perceptrón puede obtener valores de 1 y 0, es decir, una función escalonada, en cambio la función sigmoide es como un perceptrón suavizado. La suavidad de la curva de  $\sigma$  significa que para pequeños cambios  $\Delta w_j$  en los pesos y  $\Delta b$  en el sesgo producen pequeños cambios en la salida, donde podemos definir de manera aproximada como [26]:

$$\Delta salida \approx \sum_j \frac{\partial salida}{\partial w_j} \Delta w_j + \frac{\partial salida}{\partial b} \Delta b \quad (2.17.3)$$

es decir que  $\Delta salida$  es una función lineal de los cambios de  $\Delta w_j$  y  $\Delta b$ .

## 2.18. Función de costo

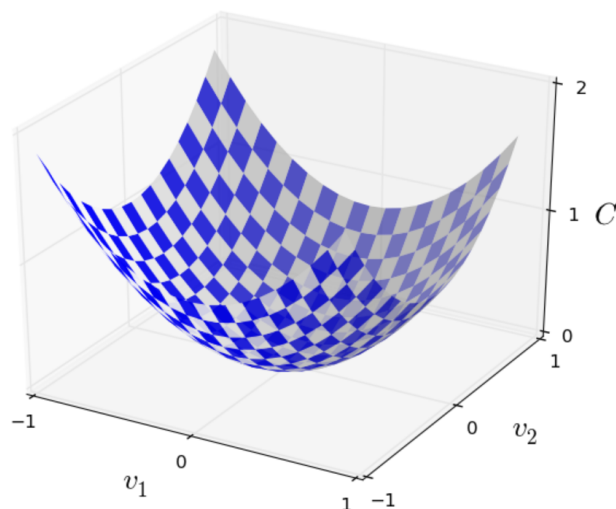
Para encontrar los pesos y sesgos correctos para un buen entrenamiento de la red neuronal, definimos la función de costo:

$$C(w, b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2 \quad (2.18.1)$$

Donde  $w$  denota la colección de los pesos de la red,  $b$  todos los sesgos,  $n$  el número de entradas de entrenamiento,  $a$  es el vector de salida de la red donde  $x$  es la entrada de la red, y la suma se aplica sobre todas las entradas para el entrenamiento. Podemos observar que  $C(w, b)$  se vuelve pequeño cuando  $y(x)$  es aproximadamente igual a la salida  $a$ , así el algoritmo debe encontrar pesos y sesgos tal que  $C(w, b) \approx 0$ , entonces el objetivo del entrenamiento es encontrar un conjunto de costos y sesgos, tal que el costo sea lo más pequeño posible. Este proceso se hará mediante un algoritmo llamado descenso de gradiente. [26]

## 2.19. Descenso de gradiente

El descenso de gradiente es un algoritmo utilizado para encontrar los valores de parámetros de una función, que minimiza una función de costo 2.18.1. En matemáticas el gradiente es una generalización multivariable de la derivada, y al igual que esta, representa la pendiente de la recta tangente a la gráfica de una función, pero a diferencia de la derivada que es una función escalar, el gradiente es una función de valor vectorial. También podemos entender al gradiente de una función escalar, evaluado en un punto generado del dominio de la función, como la dirección en la cual el campo de la función varía más rápidamente [26] [27] [28]. Dada una función  $C(k)$  con  $k = k_1, k_2, \dots$ , para poder minimizar  $C(k)$ , supongamos que  $C$  es una función de dos variables,  $k_1$  y  $k_2$ , cuya gráfica se representa como:



**Figura 2.19.1:** Gráfica de la función  $C$  de dos variables.

La idea principal es obtener donde  $C$  alcanza su mínimo global y usaremos de ejemplo la imagen 2.19.1, la cual representa una simplificación del método. Supongamos que queremos encontrar el mínimo en esta función de dos variables, por lo cual en este caso necesitamos encontrar el valle de esta función, y una forma de lograr encontrar el valle es recorrer diferentes puntos de la función y movernos en pequeñas cantidades  $\Delta v_1$  y  $\Delta v_2$  para ver la variación y poder encontrar el mínimo deseado. Este cálculo lo podremos realizar de la siguiente manera:

$$\Delta C \approx \frac{\partial C}{\partial v_1} \Delta v_1 + \frac{\partial C}{\partial v_2} \Delta v_2 \quad (2.19.1)$$

La idea principal es buscar  $\Delta v_1$  y  $\Delta v_2$  tal que  $\Delta C$  sea negativo, ya que la idea es dar con la dirección opuesta al gradiente de la función en el punto actual, porque esta es la dirección de descenso más pronunciado, y así poder encontrar un valor mínimo global, que en este caso sería el valle de la función.

De forma general el descenso de gradiente para una función multivariable  $C(Y)$  y un punto  $x$ ,  $C(x)$  disminuye más rápido si uno va del punto  $x$  hacia la dirección contraria del gradiente, es decir en la dirección del gradiente negativo, entonces tenemos que:

$$a_{n+1} = a_n - \gamma \nabla C(a_n) \quad (2.19.2)$$

con  $\gamma$  siendo la tasa de aprendizaje, donde este valor se ajusta en el proceso del entrenamiento, para que el punto  $a$  vaya en dirección del gradiente negativo en los  $n$  puntos que se recorren hasta encontrar un mínimo óptimo, así llegando a una secuencia  $C(x_0) \geq C(x_1)C(x_2)$ , tal que esta secuencia converja a un mínimo local deseado.

## 2.20. Proceso de entrenamiento de datos

El proceso de entrenamiento de una red neuronal consiste en poder llegar a valores de pesos y sesgos tal que las predicciones de la red neuronal sean precisas. Los pesos al momento de entrenar una red neuronal comienzan como valores aleatorios, a medida que la red va aprendiendo, en base a los tipos de datos de la entrada, esta ajusta los pesos mediante el sesgo, en función de cualquier error en la categorización que resultaron de los pesos anteriores a través del entrenamiento de la red.

Cuando hablamos de entrenar una red neuronal, hablamos de un proceso de

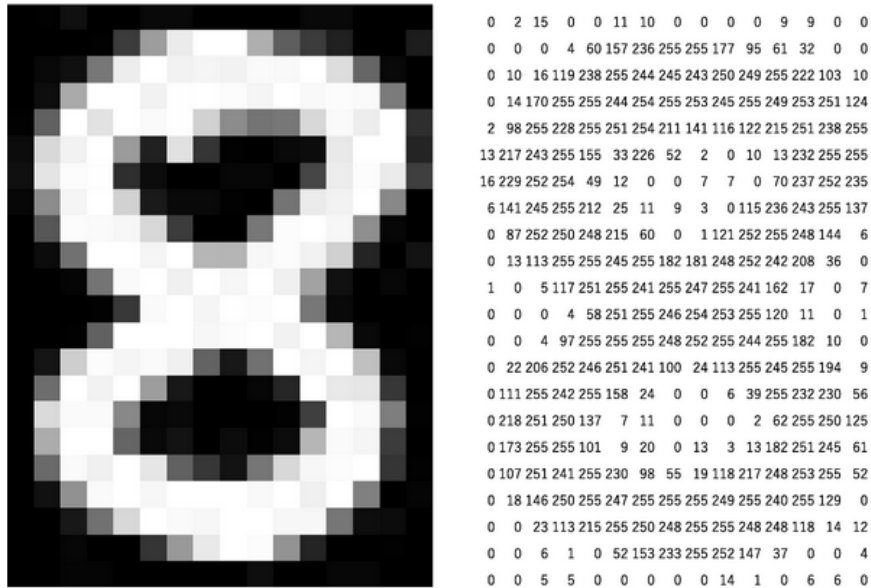
iteración de los datos etiquetados para el caso de aprendizaje supervisado. Al momento de entregar un par de datos categorizados a la red, estos pasan a través de cada capa de redes neuronales, que van captando características y obteniendo pesos y sesgos mediante diversos métodos, como pueden ser los filtros si hablamos de una imagen de entrada, y al ocurrir un primer proceso de extracción de características, la red compara ese aprendizaje con otro par de imágenes de testeo, donde la red determina que tan distantes estas los valores de sesgos y pesos respecto a los valores que obtiene de las imágenes de testeo. Una vez finalizado ese proceso, toma un valor  $\gamma$  de tasa de aprendizaje y ajusta el valor de los pesos y sesgos tal que el gradiente pueda encontrar su valor mínimo y así poder llegar a valores cercanos a lo que se va comparando con las imágenes de testeo. Este proceso ocurre por un número determinado de veces, hasta que el aprendizaje sea óptimo y se logren predicciones certeras de los datos etiquetados, este número de veces en las que se repite este proceso de entrenamiento, le llamaremos épocas.

## 2.21. Aprendizaje Profundo

El aprendizaje profundo o Deep Learning, constituye a una clase de algoritmos para el aprendizaje automático usando redes neuronales artificiales, con múltiples capas dentro de la estructura de la red, y debido a esto se asocia el adjetivo “profundo” a su nombre. Una característica que diferencia el aprendizaje profundo es la capacidad de la extracción de características. El trabajar con arquitectura y redes basadas en algoritmos de aprendizaje profundo, nos ayudará a obtener extracción de características sin un preprocesamiento a diferencia de los algoritmos planos que se ocupan en el aprendizaje automático, ya que las capas profundas de la red son capaces de aprender una representación implícita, abstracta y comprimida de los datos sin procesar [28].

### 2.21.1. Redes Neuronales Convolucionales

El objetivo principal de una red neuronal convolucional es extraer características de una imagen de entrada, donde esta imagen es representada por matrices de dos dimensiones, cuyos píxeles representa un número de la matriz con una respectiva intensidad.



**Figura 2.21.1:** Representación matricial de una imagen.

La figura 2.21.5 puede ser representada por una matriz de valores que se encuentran entre 0 y 255, donde estos valores representan la intensidad del color, siendo 0 el negro hasta 255 el blanco. La convolución se aplica directamente a la relación espacial entre píxeles, mediante el aprendizaje de las características, se compone de una o más capas convolucionales, con capas totalmente conectadas. Las redes neuronales convolucionales, intenta imitar el córtex visual del ojo humano, identificando distintas características en las entradas usando capas ocultas especializadas para las detecciones particulares, donde cada una puede detectar características básicas como líneas y curvas, y las capas profundas se encargan de reconocer características más complejas. Las redes neuronales convolucionales hacen uso de la operación matemática de convolución.

En el procesamiento de imagen, se llama kernel a una matriz utilizada para obtener características de una imagen, mediante la convolución del kernel y la matriz de la imagen. Tenemos que:

$$g(x, y) = \omega * f(x, y) \quad (2.21.1)$$

Donde  $g(x, y)$  es la imagen filtrada,  $f(x, y)$  es la imagen original y  $\omega$  el kernel del filtro.

La arquitectura básica de una red neuronal convolucional corresponde a una capa de entrada, capas ocultas y una capa de salida. Las capas ocultas suelen tener la función de aplicar distintos tipos de filtro de convolución para la abstracción

de características. Al entrenar y ajustar la red neuronal, se hace uso de la retropropagación, el cual es un algoritmo que determina el gradiente, respecto a los pesos de la red de entrada y salida, lo que genera que el cálculo sea más eficiente y no directo a cada peso de la red.

La entrada de una red neuronal convolucional requiere un tensor que contiene el número de entradas, la altura de entrada, ancho de la entrada y los canales de entrada. Al pasar por la primera capa convolucional, la entrada de la siguiente capa es el resultado de la anterior. y de esta manera sigue hasta obtener una salida en la red. Al ocurrir este proceso por cada capa, la imagen mediante el filtro de convolución va reduciendo su dimensión, es decir, la convolución reduce el número de parámetros de la red, permitiendo que la red sea mucho más profunda sin aumentar en gran cantidad el número de parámetros. Para datos con una topología similar a una cuadrícula (como las imágenes), las redes neuronales convolucionales son ideales, ya que se puede crear una relación espacial entre los parámetros mediante la convolución.

Tenemos ciertas capas convolucionales, funciones de activación y propiedades de particular interés que definir para las redes neuronales convolucionales.

#### **2.21.1.1. Sobreajuste**

El sobreajuste u overfitting es un efecto que sucede al momento de ejecutar el entrenamiento de una red. Consiste en sobreentrenar el algoritmo de aprendizaje con una gran cantidad de datos, esto produce que el algoritmo comience a captar características y aprender del ruido y de entradas de datos inexactos del conjunto de entrenamiento, por lo cual, el modelo no categoriza los datos de manera correcta, debido a los muchos detalles y ruido que captura el algoritmo de los datos. Desprendemos de que a medida que la red se entrene con sobreajuste, el error resultante de entrenamiento decrece, y la validación aumenta.

#### **2.21.1.2. Activación ReLU**

La función de activación lineal rectificadora (Rectified Linear Unit, ReLU), es función lineal, que retornara el mismo valor de entrada si este es positivo, si el valor es negativo, retornara cero.

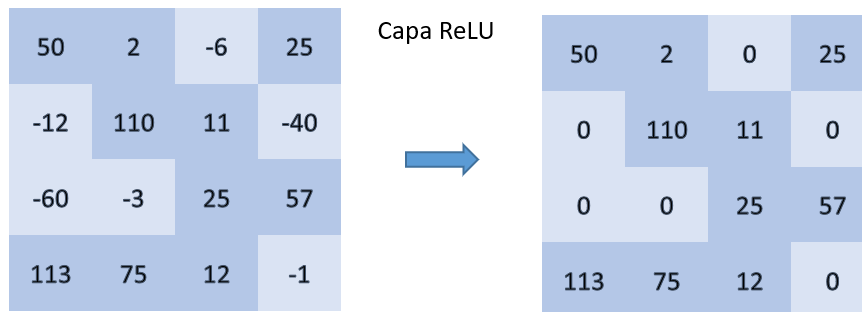
Es la función de activación más usada para la gran mayoría de redes neuronales convolucionales, por su rendimiento y facilidad a la hora de entrenar debido a su

comportamiento lineal, que genera optimización en el gradiente y se requiere de menos esfuerzo computacional, ya que ayuda a prevenir el crecimiento exponencial en el cálculo requerido para operar en las capas de la red neuronal, en comparación con usar otra función de activación como la función sigmoide 2.17.

Definimos la función de activación ReLU como:

$$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases} \quad (2.21.2)$$

Al aplicar ReLU en una capa, esta la nombraremos capa ReLU.

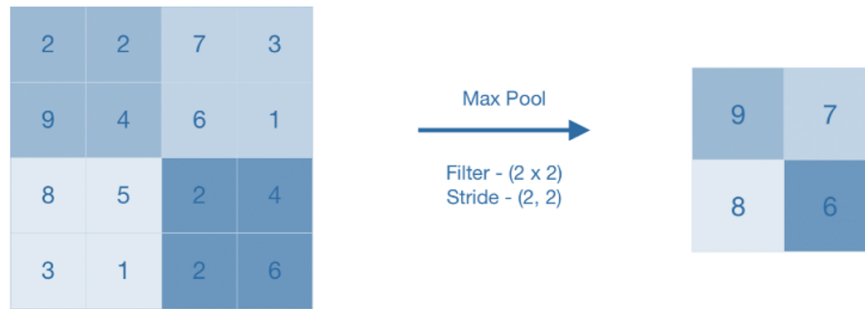


**Figura 2.21.2:** Aplicación de la capa ReLU a una matriz de característica.

### 2.21.1.3. Capa de agrupación

Las redes neuronales convolucionales pueden tener capas de agrupación locales y/o globales. La función de una capa de agrupación es reducir la dimensión, al combinar los datos generados por una capa convolucional. La capa de agrupación puede resumir las propiedades presentes obtenidas por una capa convolucional, y así, podemos reducir el número de parámetros de la red, y por lo tanto reduce la cantidad de cálculo computacional.

La capa de agrupación más usada es la capa de agrupación máxima, donde esta selecciona el elemento máximo de una región de la matriz de características, mediante un kernel asignado.



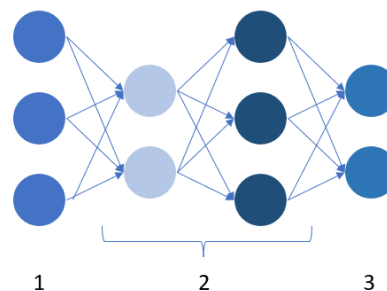
**Figura 2.21.3:** Aplicación de la Capa de Agrupación Máxima a la matriz de características.

De la figura 2.21.3 observamos que, al aplicar una capa de agrupación, se obtuvieron los valores máximos de cada stride o paso del filtro de tamaño  $2 \times 2$ , así obteniendo un mapa que contiene las características más destacadas.

También podemos usar la capa de agrupación promedio, que actúa tal cual la capa de agrupación máxima, pero obtenemos un promedio de los valores y no el máximo al aplicar el filtro.

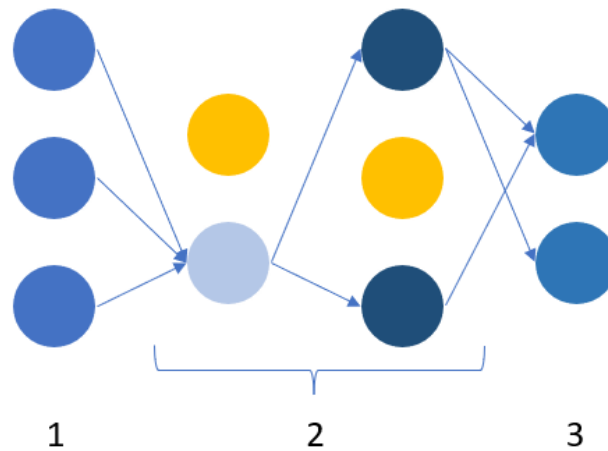
#### 2.21.1.4. Capa de abandono

La capa de abandono o capa Dropout, tiene la funcionalidad de eliminar de forma aleatoria el aporte de algunas neuronas hacia la siguiente capa, transformando un valor aleatorio en cero. Este es un método de regularización muy barato computacionalmente y notablemente efectivo para reducir el sobreajuste y mejorar el error de generalización en redes neuronales profundas. El objetivo es que cada capa se vea y se trate diferente a la capa anterior y así, independiente de que los datos tengan mucha similitud, habrá cambios aleatorios que ayudarán a tener variaciones.



**Figura 2.21.4:** Esquema de una red neuronal convolucional. En 1 tenemos las capas de entradas, en 2 tenemos las capas ocultas y en 3 la capa de salida.

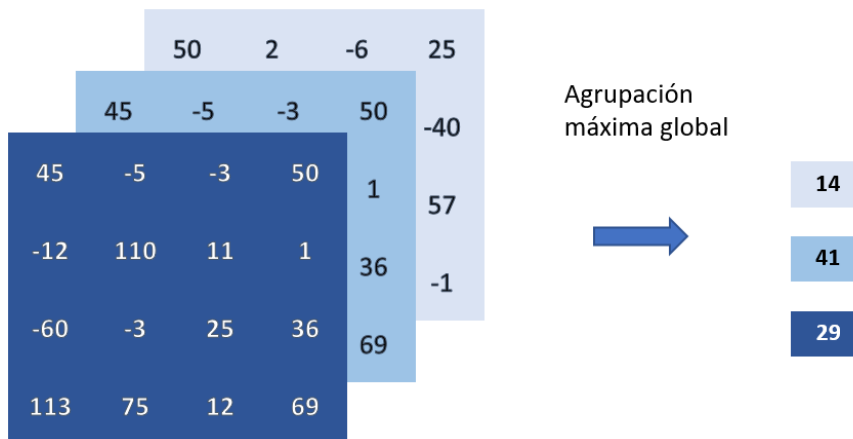




**Figura 2.21.5:** Esquema de una red neuronal convolucional aplicando Dropout a las capas ocultas.

#### 2.21.1.5. Agrupación media global

Aplicando una agrupación media global a los mapas de característica de la última capa convolucional, la cual es totalmente conectada, se obtiene un vector resultante de los promedios de cada capa de características.

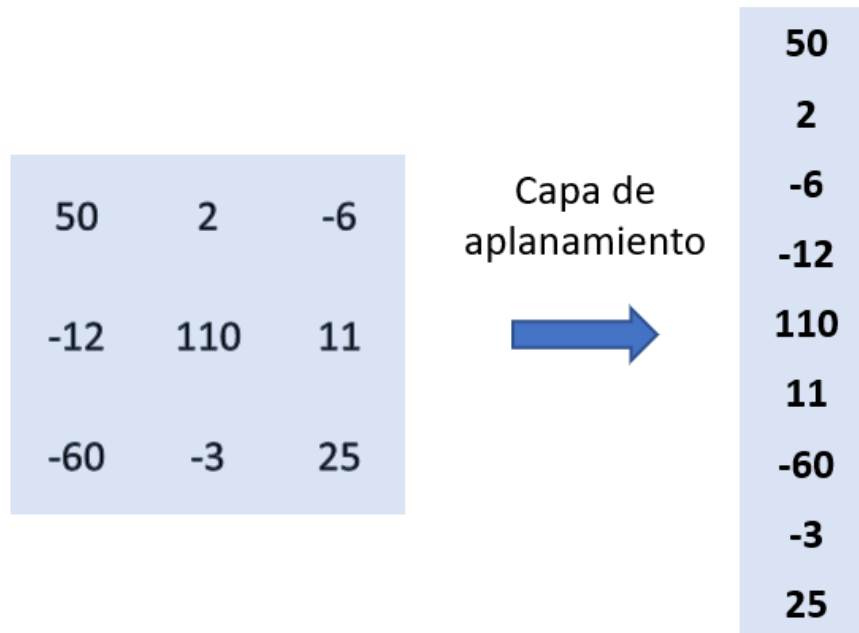


**Figura 2.21.6:** Esquema de funcionamiento de la capa de agrupación media global.

En la figura 2.21.7 tenemos en 1, los mapas de características de la última capa convolucional que al aplicar la agrupación media global, obtenemos un vector cuyos elementos es el promedio asociado a cada capa.

### 2.21.1.6. Capa Plana

La Capa plana, aplanar la salida de una capa a un vector de una dimensión. Esta se suele usar como transición entre una capa convolucional y la Capa Densa, la que definiremos en la siguiente sección.



**Figura 2.21.7:** Esquema de funcionamiento de la capa de aplanamiento.

### 2.21.1.7. Capa Densa

La capa densa implementa la operación:

$$salida = \sum w_i \cdot x_i + b \quad (2.21.3)$$

Donde  $x_i$  son los valores de la capa de entrada,  $w_i$  los pesos de la red neuronal convolucional y  $b$  el sesgo. La salida de la capa densa, es una combinación lineal del producto punto de valores de entrada y los pesos, sumados al sesgo.

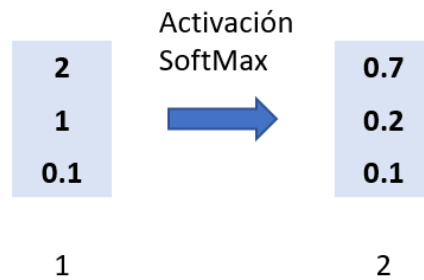
### 2.21.1.8. Función SoftMax.

La función softmax o función exponencial normalizada, se aplica a un vector K-dimensional  $x$ , para comprimir a un vector K-dimensional  $\sigma(x)$  de valores reales en el rango de  $[0, 1]$ .

Se define como:

$$\sigma(x_i) = \frac{e^{x_i}}{\sum_{k=1}^K e^{x_j}} \quad (2.21.4)$$

con  $i = 0, 1, 2, \dots, K$ .



**Figura 2.21.8:** Esquema de funcionamiento al aplicar la función de activación SoftMax. En 1 tenemos la capa de entrada y en 2 la capa de salida.

En la figura 2.21.8 tenemos que los valores de 1 provienen de una capa anterior, los cuales se transformarán a valores de porcentaje en 2, a través de la activación softmax.

La función SoftMax es utilizada como activación en la capa final de redes neuronales basados en la clasificación. Se suele usar como función de activación en la capa Densa 2.21.1.7, donde actúa sobre la entrada producida por la ecuación 2.21.3 que a su vez recibe una capa de una dimensión producida por la capa plana, obteniendo un porcentaje de clasificación en la salida de la capa Densa, con activación SoftMax.

## Capítulo 3

# Metodología

El objetivo principal de este trabajo es lograr desarrollar un sonómetro, que sea capaz de tener todas las características principales de un sonómetro estándar de alta calidad, agregando un componente de aprendizaje automático de reconocimiento de sonido, y así poder entrenar un modelo de redes neuronales bajo contextos de entornos sonoros específicos, que nos ayudaran a poder relacionar la intensidad de ruido con un evento en específico.

El enfoque principal en el desarrollo de este instrumento, es utilizar productos de bajo costo que logren equiparar la funcionalidad de sonómetros de alta calidad, como los sonómetros de clase 1 y 2 [11], cuyo valor monetario es significativo respecto a los equipos usados para desarrollar este trabajo.

El desarrollo de este instrumento se hizo en conjunto con la empresa SoundAnalytics [12], que ofrece metodologías de análisis en tiempo real de entornos sonoros a diferentes entidades para la evaluación de la norma de regulación de la contaminación acústica en Chile [10]. Esta norma define rangos de ruido máximo en decibeles durante el día y noche para diferentes zonas, en particular SoundAnalytics [12], estudia y analiza el impacto del ruido que generan industrias y proyectos de desarrollo en la salud de la población.

<b>Ruido máximo en decibeles (dB) durante el día</b>		
	De 7 a 21 Hrs.	De 21 a 7 Hrs.
Zona I	55	45
Zona II	60	45
Zona III	65	50
Zona IV	70	70

**Cuadro 3.0.1:** Norma de Regulación de la contaminación acústica en Chile.

Donde cada zona se define como:

- **Zona I:** Aquella zona definida en el Instrumento de Planificación Territorial respectivo y ubicada dentro del límite urbano, que permite exclusivamente uso de suelo Residencial o bien este uso de suelo y alguno de los siguientes usos de suelo: Espacio Público y/o Área Verde.
- **Zona II:** Aquella zona definida en el Instrumento de Planificación Territorial respectivo y ubicada dentro del límite urbano, que permite además de los usos de suelo de la Zona I, Equipamiento de cualquier escala.
- **Zona III:** Aquella zona definida en el Instrumento de Planificación Territorial respectivo y ubicada dentro del límite urbano, que permite además de los usos de suelo de la Zona II, Actividades Productivas y/o de Infraestructura.
- **Zona IV:** Aquella zona definida en el instrumento de Planificación Territorial respectivo y ubicada dentro del límite urbano, que permite sólo usos de suelo de Actividades Productivas y/o de Infraestructura.

Uno de los objetivos principales de SoundAnalytics [12] es registrar, medir, analizar y elaborar planes de acción para que las industrias y proyectos de desarrollo cumplan con estándares y normativas como la mencionada en la tabla 3.0.1.

Los valores de decibeles en las mediciones de nivel de presión sonora que se realizan en cada zona, son mediciones promediadas por minuto en *dBA*. Teniendo en cuenta estas especificaciones, se desarrollará un sonómetro que entregue información bajo esta normativa.

## 3.1. Objetivo de medición.

El sonómetro será capaz de realizar:

1. Mediciones por segundo en *LASeq* y *LZSeq* en tercios de octavas.
2. Valores máximos y mínimos de medición para cada filtro de ponderación.
3. Ejecución de modelo de Aprendizaje automático segundo a segundo.
4. Almacenamiento de datos clasificados para entrenar una red neuronal.
5. Almacenamiento de audio continuo en  $44,1kHz$ .
6. Acceso remoto.
7. Respaldo y almacenamiento de datos en la nube.
8. Interfaz gráfica y web para interactuar con los datos.

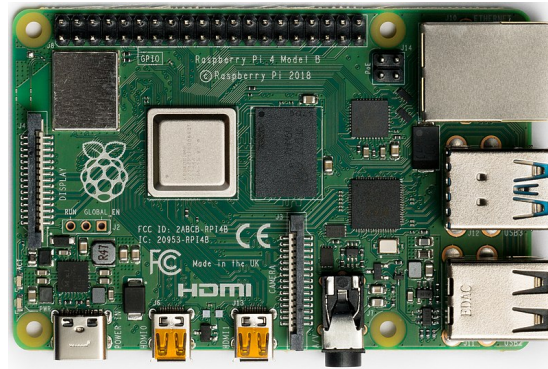
Estas funciones se aplicarán respectivamente a dos tipos de sonómetros desarrollados. El sonómetro de mano y el sonómetro continuo.

Para el desarrollo del sonómetro, se utilizará un ordenador de placa simple que ejecutará un sistema operativo, donde se usarán diferentes códigos de programación, los que cumplirán con la función de realizar las tareas de un sonómetro y de ejecutar el modelo de aprendizaje automático. Para la captura de audio, se le acoplará al ordenador de placa simple, una tarjeta de sonido USB conectada a un micrófono condensador omnidireccional.

## 3.2. Hardware.

### 3.2.1. Raspberry pi.

El sonómetro se desarrolló en un ordenador de placa simple, creados por Raspberry Pi Foundation [29]. El modelo usado fue un Raspberry Pi 4 modelo B [30] con 8 GB de memoria RAM, con un procesador ARM Cortex-172 con cuatro núcleos a 1,5 GHz.

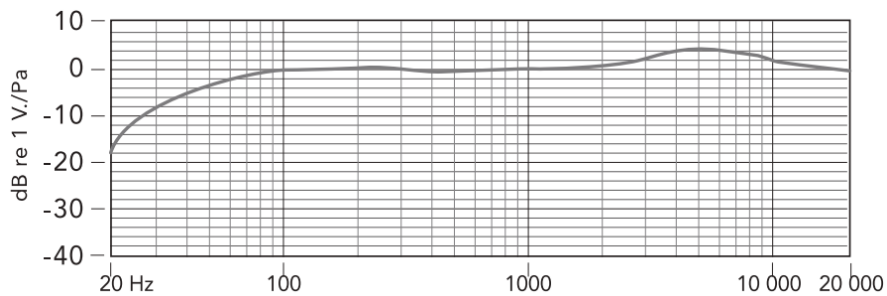


**Figura 3.2.1:** Raspberry pi modelo 4 B+.

Modelo 4 B con 2 X USB 2.0 y 2 X USB 3.0, 2 x micro-HDMI, una entrada ethernet, entrada de tarjeta microSD, entrada de corriente 3 A- 5V y 2x20 pin GPIOs.

### 3.2.2. Micrófono.

El micrófono utilizado es el BOYA BY-M1 [31], un micrófono con patrón polar omni-direccional, es decir, que captura sonido en todas direcciones, con conector de 3.5 mm.



**Figura 3.2.2:** Respuesta de frecuencia del modelo BY-M1.



**Figura 3.2.3:** Micrófono BOYA BY-M1.

### 3.2.3. Tarjeta de sonido USB.

La tarjeta de sonido USB utilizada es de la marca VENTION modelo CDLH0 [32]. Cuenta con un puerto auxiliar de 3.5 mm, entrada USB 2.0, frecuencia de muestreo de 44,1/48K@16bit.

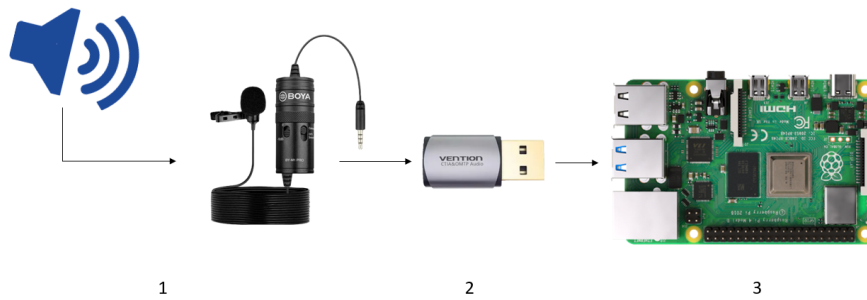
La tarjeta de sonido tiene la función de conectar la placa del ordenador al micrófono, procesando la señal analógica del micrófono en información de audio digital.



**Figura 3.2.4:** Tarjeta de sonido USB Vention - CDLH0.

El micrófono y la tarjeta de sonido USB, pueden ser de cualquier otro modelo o marca que cumpla con las mismas características, no es de uso exclusivo para el funcionamiento del sonómetro, pero existen ajustes específicos para cada micrófono e interfaz de audio USB como lo es la calibración ambiental, proceso que será desarrollado más adelante.





**Figura 3.2.5:** Esquema del hardware.

En la figura 3.2.5 podemos ver el diagrama de funcionamiento del hardware, en 1 ocurre el proceso de captura de audio mediante el micrófono, en 2 la interfaz de audio USB procesa la señal analógica del micrófono y la convierte en información de audio digital para que, en 3 la Raspberry pi a través del sistema programado, sea capaz de procesar el audio.

### 3.3. Sistema Operativo.

El sistema operativo que se ejecutaron en Raspberry pi 4, es el Raspberry Pi OS (64-bit), distribución del sistema operativo GNU/Linux basado en Debian, siendo este creado específicamente para las SBC Raspberry pi, teniendo así total compatibilidad, altamente optimizado y estable. Este sistema operativo será el encargado de procesar y ejecutar los códigos de programación encargados de las funcionalidades del sonómetro, procesando el audio digital dado por la interfaz de audio.

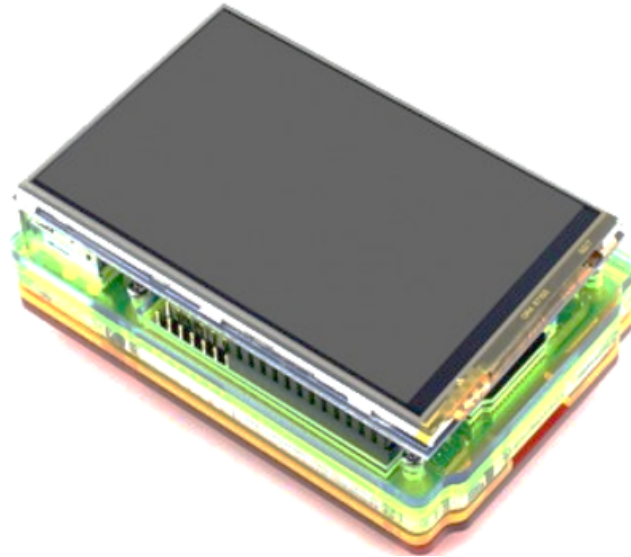
### 3.4. Sonómetro de mano.

#### 3.4.1. Hardware

El sonómetro de mano tiene como característica ser transportable y desplegar información en pantalla.

Se añadió a la Raspberry pi 4 una pantalla táctil que se conecta a través de las entradas GPIOs. Esta pantalla permitirá desplegar la información que captura el sonómetro a través de una interfaz gráfica, contando también con la interfaz de

audio USB y micrófono.



**Figura 3.4.1:** Raspberry pi con pantalla táctil LCD.

### 3.4.2. Procesamiento de datos.

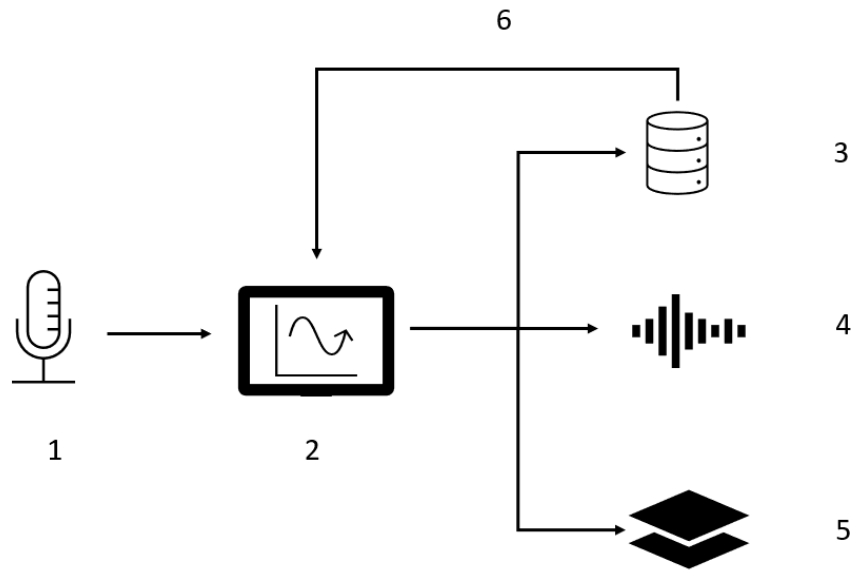
Para poder procesar el audio digital que captura la interfaz de audio USB, se crea un programa realizado en el lenguaje de programación Python 3.10 [33], usando el módulo Sounddevice [34], que crea enlaces para la librería PortAudio [35], que permite interactuar con los datos entrantes de la interfaz de audio USB y nuestro programa. Se define un flujo de datos de audio entrante a  $44,1\text{KHz}$  con un solo canal. Este flujo de datos se lee segundo a segundo, seccionando el flujo de datos en bloques de 44100 datos por segundo. Con los datos entrantes, calculamos la transformada de Fourier para obtener el espectro de frecuencia, usando el algoritmo de FFT mediante la librería Numpy [36], luego calculamos el nivel de presión sonora en tercios de octavas, aplicamos la ponderación temporal obtenemos *LZSeq* para cada una de las bandas de tercios de octavas. Aplicando el filtro de ponderación *A* obtenemos de igual manera *LASeq*.

A medida que se calculan estos datos segundo a segundo, se van almacenando en una base de datos relacional SQLite [37], con la fecha y hora correspondiente a cada segundo del registro. Al mismo tiempo, se va almacenando por segundo un archivo de audio que contiene todo el registro. De igual manera, por segundo se ejecuta un modelo de Aprendizaje automático llamado YAMNet [38]. Este modelo

contiene 521 categorías para la predicción de audio, entrenadas con el corpus AudioSet-Youtube [39] [40], el cual fue entrenado con más de 2.1 millones de audios contenidos en vídeos de Youtube, teniendo así más de 5800 horas de audio. Este modelo usa la arquitectura Mobilenet\_v1 [41], arquitectura optimizada para dispositivos de poca potencia computacional. El modelo se modificó para el sonómetro, quedando con una rápida ejecución, ya que el modelo original, el cual requiere mucho procesamiento de ejecución, se transformó a un modelo optimizado específicamente para este sonómetro. La ejecución de este modelo se realiza mediante la librería TensorFlow [42], y en particular usando su librería TensorFlow Lite. TFLite es un conjunto de herramientas que nos permite ajustar modelos para incorporarlo a dispositivos de bajo procesamiento teniendo en cuenta la optimización, baja latencia, con un eficiente consumo de energía y sin necesidad de estar conectado a un servidor ya que la ejecución es en el dispositivo, lo que genera privacidad en los archivos.

Este modelo, al ser entrenado con millones de datos de audios nos servirá como una primera instancia para dar un contexto al entorno sonoro que captura el sonómetro. Si necesitamos saber a qué corresponden ciertos niveles de presión sonora en el registro de los audios, podemos acceder a los datos de predicciones y saber qué fue lo que registró en un segundo determinado. El sonómetro es capaz de almacenar en 521 carpetas, muestras de un segundo de audio con la clasificación correspondiente, lo que nos permitirá poder crear nuestra propia base de datos para la creación de una red neuronal profunda bajo un contexto específico de entorno sonoro. Las muestras de audios son almacenadas siempre que superen el 70 % de exactitud en la predicción.

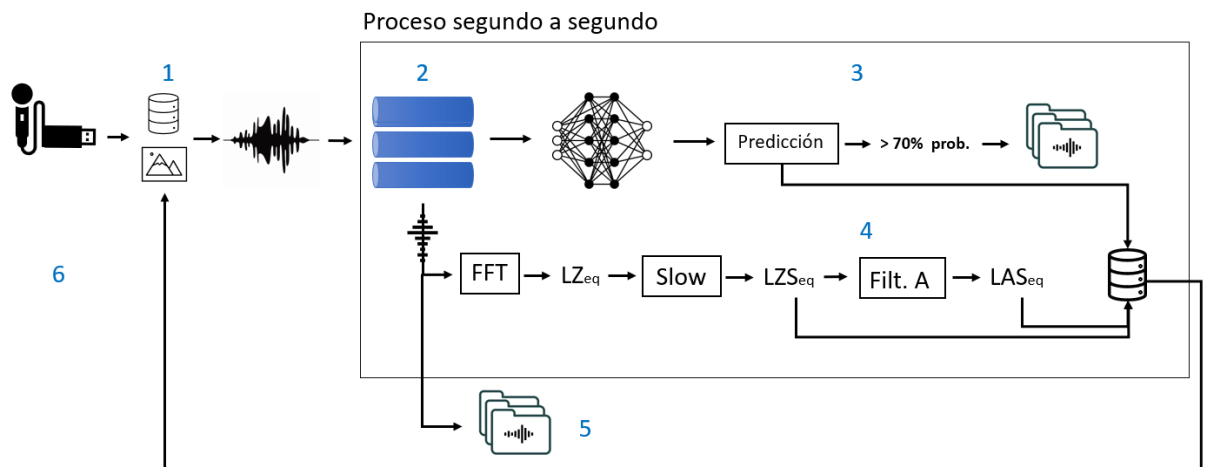
Todas estas funcionalidades son mostradas en pantalla a través de una interfaz gráfica desarrollada con Tkinter [43] con la que se puede interactuar a través de tres pestañas. La pestaña 1 nos mostrará los gráficos en tiempo real con nivel de medición de segundo a segundo. La segunda pestaña nos mostrará datos específicos de los niveles de medición y la tercera pestaña las clasificaciones que se ejecutan en tiempo real.



**Figura 3.4.2:** Diagrama de funcionamiento del Sonómetro de mano.

En la figura 3.4.2 se describen los distintos procesos que ejecuta el sonómetro de mano. En 1 ocurre el proceso de captura de sonido ejecutado por el micrófono y la interfaz de audio USB, en 2 ocurre el procesamiento de audio, para luego en 3 guardar la información calculada en la base de datos, en 4 guardar el audio del registro y en 5 se almacena las predicciones del modelo y audios en distintas carpetas según la clasificación. En el proceso 6, la información almacenada en la base de datos es enviada a la interfaz gráfica para desplegar la información.

Para el proceso interno de 2 tenemos:



**Figura 3.4.3:** Diagrama de funcionamiento completo del sonómetro de mano.

En 1, al recibir la información analógica capturada por el micrófono y transformada a señal digital por la interfaz de audio USB, nuestro programa inicializará la interfaz gráfica y crea una base de datos nueva para cada medición. Luego en 2 comienza el proceso que se ejecuta segundo a segundo. El programa crea colas (Queue), que son estructura de datos lineal que almacenan de manera óptima nuestros datos, teniendo la escritura FIFO, es decir, primer dato que entra es el primero que sale al ser llamado. Estas colas vacías son las que guardaran nuestros datos entrantes y también procesados. En 3 tenemos la ejecución del modelo clasificador de sonido. Este modelo guarda en la base de datos todas las predicciones y almacena el audio de las predicciones en las respectivas carpetas cuando las predicciones tienen un 70 % o más de exactitud. En 4 se calcula *LZSeq* y *LASeq*, donde estos datos y los contenidos en cada banda de tercio de octava se almacenan en la base de datos. En 5 se guarda el audio completo del registro y por último, en 6, todo lo guardado en el segundo se despliega en pantalla en la interfaz gráfica, actualizando así los datos por segundo. Este proceso se ejecuta hasta que el usuario presiona un botón en la pantalla táctil que detendrá el proceso de registro.

Se desarrollo otra aplicación integrada al sonómetro, que lista los registros completos guardados. Al colocar un pendrive USB, la aplicación lo leerá y montará en el sistema, luego el usuario debe seleccionar el botón de transferencia de datos y en el pendrive se crearán 4 carpetas, una con los audios de los registros, una con las bases de datos, otra con un archivo XLSX (Excel) que muestra los registros de la base de datos y por último una con los audios clasificador por el modelo.

### 3.5. Sonómetro Continuo.

El sonómetro continuo a diferencia del sonómetro de mano tiene como función principal ser ubicado en una posición fija, donde registre de forma continua durante periodos prolongados de tiempo, ya sean semanas o meses la información del entorno sonoro. Este sonómetro no cuenta con pantalla debido a que es posible acceder y configurar de forma remota, sube información a la nube e interactúa los datos con una aplicación Web. Este sonómetro fue desarrollado para SoundAnalytics, por lo que su diseño y función, se enfoca principalmente en los descriptores de la norma [10], entonces el objetivo es tener niveles de *LASeq* por minuto y además desplegar la información en una aplicación web.

### 3.5.1. Hardware

Este tipo de sonómetro, al funcionar de forma continua durante grandes periodos de tiempo, está construido de manera de que su comportamiento sea el óptimo y eficiente a largo plazo. Este cuenta con una carcasa específica para el modelo de Raspberry pi 4, ventiladores conectados a las entradas GPIOs, disipadores y una caja contenedora que lo protege del exterior. También cuenta con una banda ancha móvil (BAM) que permite conectar el sonómetro a internet.

### 3.5.2. Funcionamiento

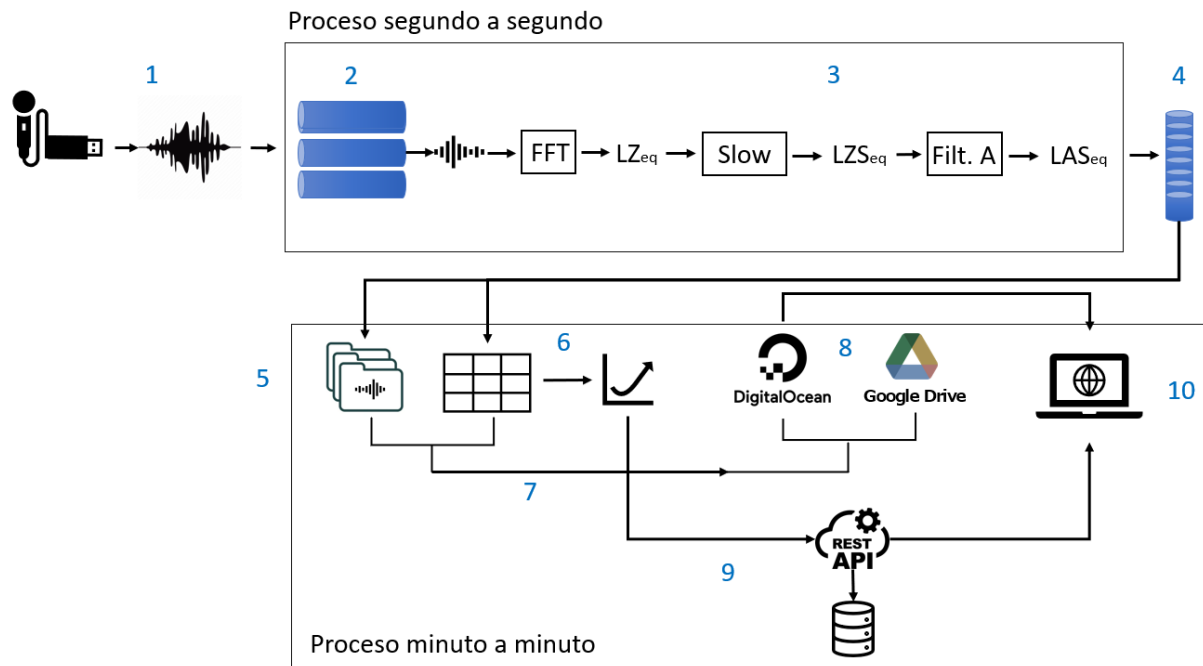
Al ser un dispositivo continuo que está comunicado con una aplicación web, está conectado a corriente directa de forma constante y a una banda ancha móvil para comunicarse y enviar los datos a la aplicación web. El sonómetro fue configurado para ejecutar todas sus funciones de forma automática una vez que recibe corriente. Esta medida es debido a los posibles cortes de corriente y para la facilidad de manejo del dispositivo. Por otro lado, al estar conectado siempre a internet mediante la banda ancha móvil, es posible poder controlar el dispositivo de forma remota, para realizar monitoreo y aplicar actualizaciones.

### 3.5.3. Procesamiento de datos.

Al ser de funcionamiento continuo, el tratamiento del procesamiento de datos es diferente al que se aborda con el sonómetro de mano. Para el cálculo de *LASeq* por minuto se implementa el mismo algoritmo de funcionamiento. Dada una señal digital entrante, se realiza el cálculo de *LASeq* segundo a segundo mediante la FFT 2.7, calculando el nivel de presión sonora equivalente, aplicando filtro ponderado A y ponderación temporal. Se guarda un archivo la información minuto a minuto de *LASeq* y también un archivo de audio. Sobre este archivo que contiene los datos del minuto, se calcula el promedio de *LASeq* y este es enviado mediante una solicitud a una APIRest [44] (interfaz utilizadas por dos sistemas computacionales para intercambiar información de manera segura a través de Internet), cuya tarea es recibir datos entrantes y desplegarlo en la página web de SoundAnalytics [12]. Los archivos de datos y audio son respaldados en servidores en la nube tales como DigitalOcean [45], que es un proveedor de servidores virtuales privados, y

en Google Drive. Los audios almacenados en DigitalOcean son solicitados por la página web de SoundAnalytics [12] para las gráficas interactivas.

Como medida adicional, se tiene un monitoreo minutal de temperatura, factor de gran importancia en dispositivos que están en exteriores y que pueden alcanzar altas temperaturas. Esto permite accionar y mejorar el contenedor del dispositivo para obtener unas condiciones mas óptimas para su funcionamiento. El funcionamiento descrito del sonómetro es un proceso que se ejecuta minuto a minuto.



**Figura 3.5.1:** Diagrama de funcionamiento de sonómetro continuo.

En el diagrama de funcionamiento del sonómetro continuo 3.5.3 tenemos en 1 al igual que en el sonómetro de mano, la captura de sonido mediante el micrófono y la interfaz de audio USB, al inicializar la captura de audio se crea una cola que almacenará el flujo de audio entrante. Luego en 3 se aplica el algoritmo de cálculo de *LASeq* que es almacenado segundo a segundo en una cola (4) hasta que se complete un minuto. Luego de ser almacenado los datos de *LASeq* del minuto, esta información es almacenada como audio y archivo de datos, como se muestra en 5, luego el archivo de datos es procesado (6) y así se obtiene un promedio de *LASeq* para el minuto. El audio y los datos almacenados son respaldados en DigitalOcean y Google drive (7). El promedio calculado es enviado a la API Rest (9), que almacena el promedio en una base de datos y envía los datos para que

puedan ser desplegados en la aplicación web (10), donde esta también es capaz de reproducir los audios obteniéndolos desde DigitalOcean. Mientras ocurre el proceso minuto a minuto del envío de datos a la aplicación web, la ejecución del cálculo de *LASeq* continua.

#### 3.5.4. Red Neuronal Profunda

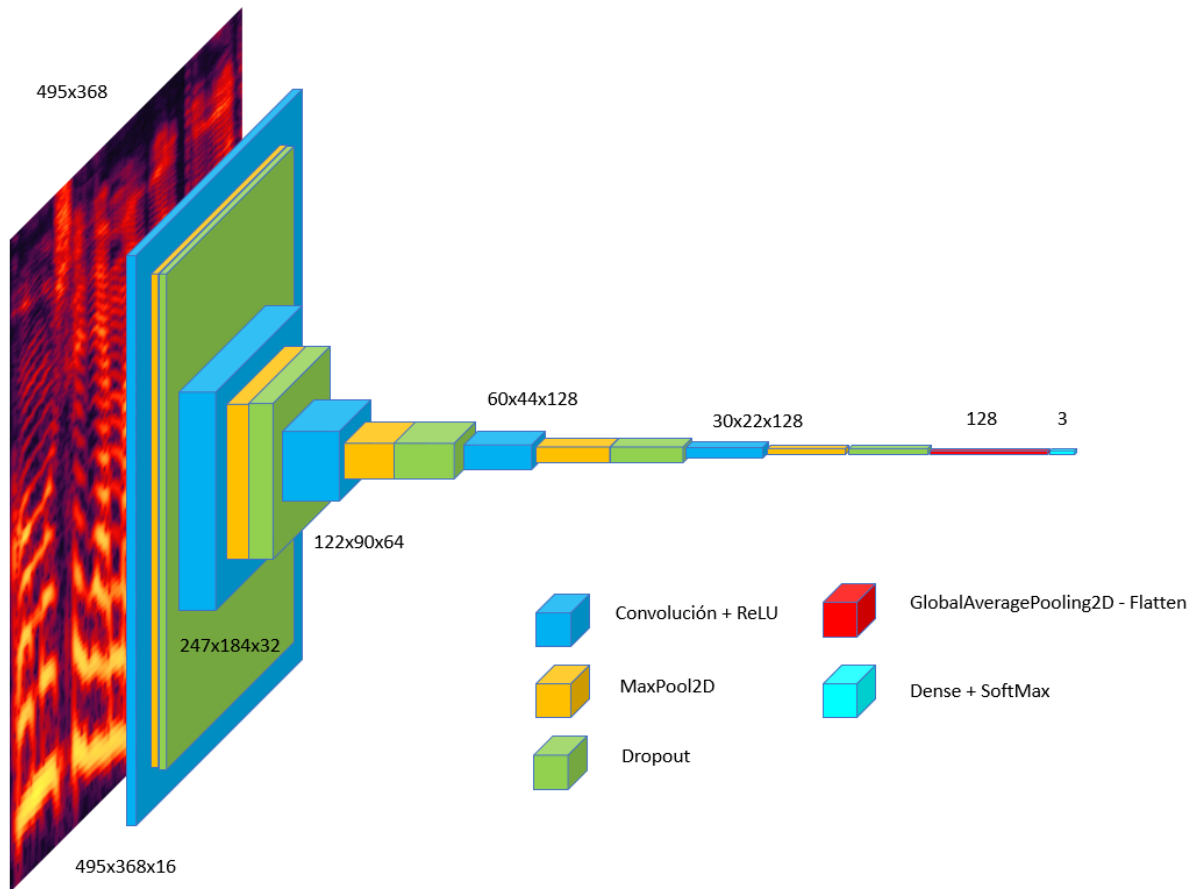
En el proceso de registros de mediciones en una de las empresas en la que opera SoundAnalytics, se creó una red neuronal profunda para la detección de eventos en un entorno sonoro en particular, en donde interactúan población, fauna e industria. Este entorno sonoro tiene eventos de sonido de industria importantes para el análisis y los que son importantes de detectar, debido a que estos generaban un nivel muy alto en decibeles, sin embargo eran difícil diferenciar cuando los promedios de *LASeq* subían respecto a esos eventos o a elementos del entorno sonoro como la naturaleza o población urbana. Por lo tanto, el implementar una red neuronal profunda permitiría saber que eventos ocurren segundo a segundo y así poder diferenciar si el aporte de decibeles al entorno sonoro era producto de la industria o de la población y/o fauna.

Se definieron en principio 8 categorías de datos, cuyo aporte era de mayor nivel de ruido en general, las cuales eran: Viento, Aves, canto del Vallenus Chilensis [20], cadenas de la industria, caídas de tronco de la planta, fondo de la planta, Carretera y fondo externo a la planta. Teniendo en cuenta estas categorías, en un principio se comenzó a recolectar de forma manual archivos de audios que tuviera una muestra de un segundo de ellas, de esa manera se creó una base de datos con 2798 archivos de sonido entre las 8 categorías para el entrenamiento de una red neuronal profunda.

La base de datos fue procesada y transformada a muestras de un segundo para cada categoría, estas muestras fueron transformadas a una matriz que contenía el espectrograma del fragmento de audio, en virtud de ello, se utilizó una red neuronal convolucional para poder capturar la relación espacial entre los parámetros mediante la convolución y crear las predicciones en base a esa matriz. La matriz del espectrograma tiene una dimensión de  $495 \times 368$  para la muestra de un segundo de audio a  $44,1\text{KHz}$ .

La red neuronal profunda tiene una arquitectura de la forma:





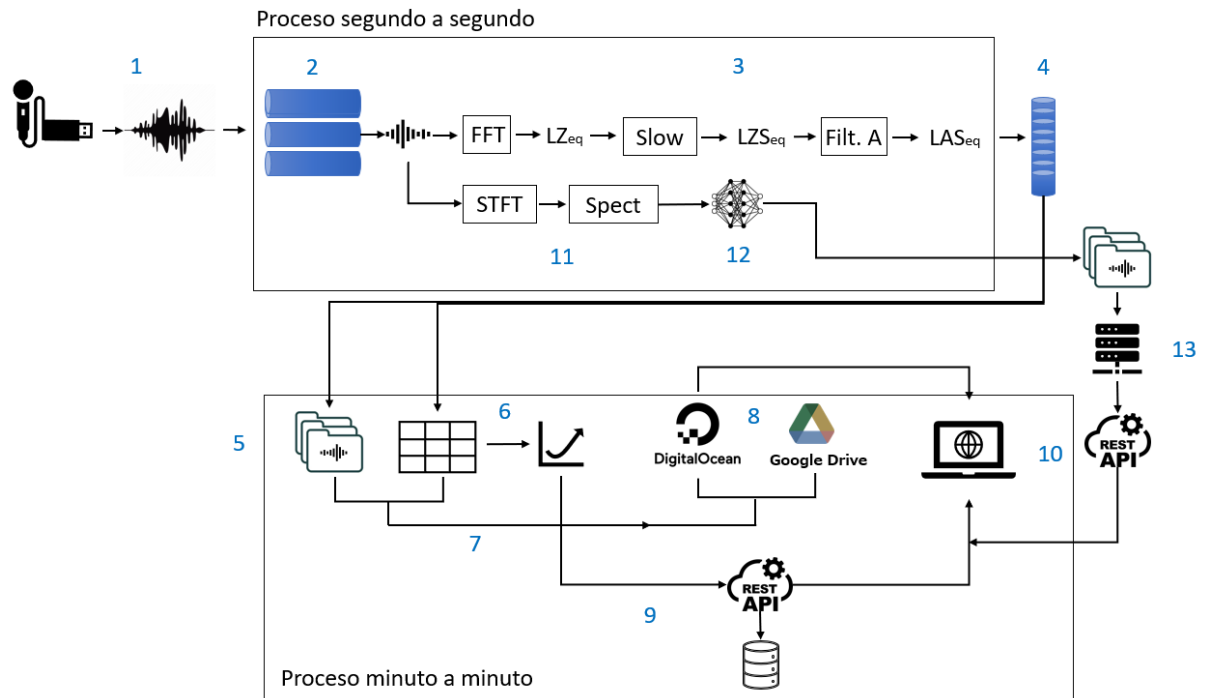
**Figura 3.5.2:** Arquitectura de Red Neuronal profunda.

La entrada del modelo requiere una matriz de 498x368, que corresponderá a la matriz del espectrograma de un segundo de audio, debido a esto, antes de ejecutar el modelo se ejecuta un preprocesamiento que calcula el espectrograma. Mediante la ecuación 2.9.1 podemos calcular el espectrograma.

El modelo fue transformado para funcionar en dispositivos de baja capacidad computacional, teniendo latencia óptima, privacidad en los datos y un tamaño reducido, transformado flotantes de 32 bits en enteros de 8 bits más eficientes.

La salida del modelo mediante la activación SoftMax nos entrega un vector de 3 elementos con un valor de precisión asociado a cada clasificación.

La ejecución del modelo, según el diagrama de funcionamiento 3.5.3, se aplica al igual que el cálculo de *LASeq* segundo a segundo. Las predicciones son enviadas a un servidor y mediante una APIRest son graficadas en la aplicación web. La gráfica muestra un porcentaje de cuantas veces se detectaron las categorías dentro de un minuto de grabación.



**Figura 3.5.3:** Diagrama de funcionamiento de sonómetro continuo con Modelo de aprendizaje automático.

El sonómetro continuo integrado con el modelo de aprendizaje profundo, agrega el proceso 11 donde se calcula el espectrograma de un segundo de audio, luego este es introducido al modelo, que ejecuta y retorna una predicción 12. Después de un minuto se almacenan todas las predicciones en un archivo y son enviadas al servidor, el cual envía mediante una APIRest los datos a la aplicación web.

Al tener la red entrenada, que tenga una buena precisión y retorne datos confiables, será la encargada de detectar, clasificar nuevos eventos en los audios para aumentar la base de datos y así reentrenar el modelo. Entonces entramos en un ciclo de retroalimentación, en el que el modelo es capaz de encontrar nuevos datos para el reentrenamiento y así, aumentar la capacidad de detección del modelo.

### 3.6. Homologación.

El objetivo de ambos sonómetros 3.1 es cumplir con ciertas características de un sonómetro de alta gama, que están regulados bajo normativas internacionales como IEC 61672-1:2013 [11] que nos garantizan un funcionamiento con criterios de precisión, rendimiento y calibración que los instrumentos deben cumplir.

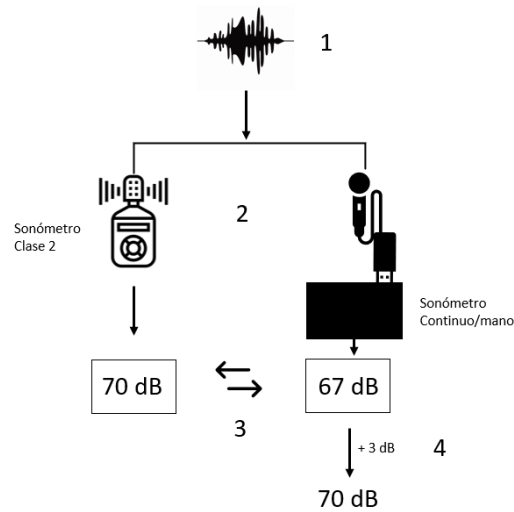
Los dos tipos de sonómetros fueron calibrados ambientalmente, proceso que llamaremos homologación, respecto a niveles de medición de un sonómetro clase 1. El sonómetro usado de referencia es el Larson Davis Modelo SoundTrack LXT. [46]



**Figura 3.6.1:** Sonómetro Larson Davis Modelo SoundTrack LXT.

El proceso de homologación del sonómetro continuo y de mano, era mediante el registro simultaneo en entorno sonoro durante largos periodos de tiempo, para luego poder comparar los niveles de *LZSeq* y *LASeq* segundo a segundo y entre el sonómetro clase 2 y los sonómetros fabricados, para poder corregir la diferencia que existe respecto al sonómetro clase 2 en los niveles de presión sonora. También se calcularon promedios minutales para analizar la diferencia con respecto a la referencia.

Este proceso se realizó usando el hardware específico, por lo que para otro micrófono o interfaz la homologación es diferente, ya que la respuesta al sonido puede variar.



**Figura 3.6.2:** Diagrama de funcionamiento de homologación.

El proceso de homologación lo podemos ejemplificar con el diagrama de la figura 3.6.2. Comienzan a registrar el sonómetro clase 2 y el sonómetro continuo o de mano el mismo entorno sonoro durante un rango de horas (1). Ambos sonómetros luego del registro guardan la información capturada en sus respectivos archivos (2). Cada uno de esos niveles capturados segundo a segundo se comparan (3). Usando como referencia el sonómetro de clase dos, se ajusta el cálculo del sonómetro continuo o de mano para alcanzar los mismos niveles (4) a través de un promedio de las diferencias obtenidas entre cada segundo, obteniendo así un parámetro de homologación. Este parámetro se agrega a la configuración interna del sonómetro continuo o de mano. Para verificar su funcionamiento y precisión, se calculan promedios minutales y se compara la diferencia obtenida ahora con el sonómetro homologado respecto al sonómetro de clase 2 en un nuevo registro.

# Capítulo 4

## Análisis

### 4.1. Sonómetro continuo.

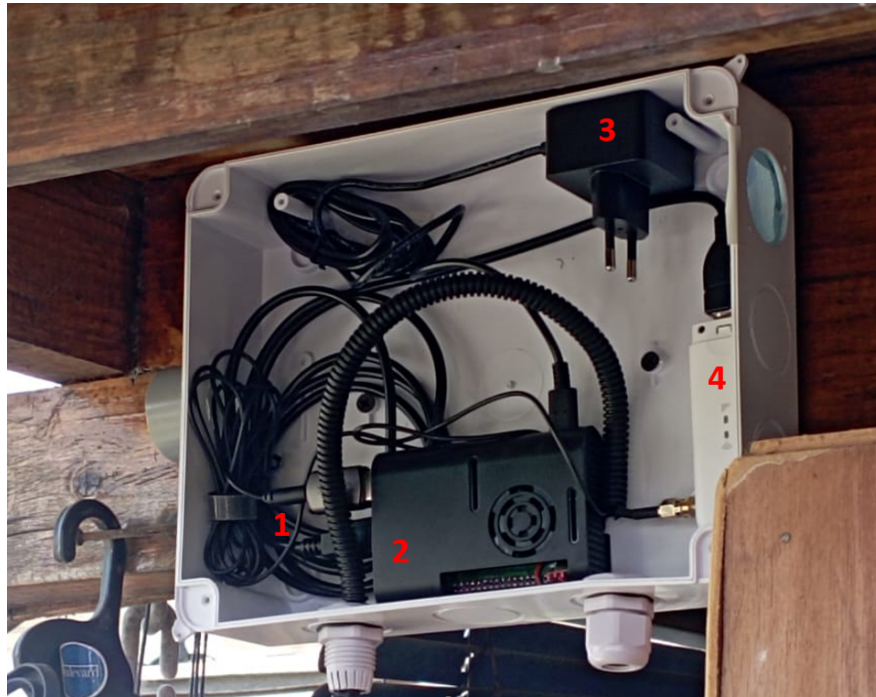
#### 4.1.1. Montaje

El sonómetro continuo cuenta con las siguientes especificaciones:

Sonómetro continuo.	
<b>Mediciones</b>	
Filtros de medición (dB SPL)	A Z
Tipos de mediciones	SPL, LZSeq, LASeq, LASeq promedio, Audio de registro, Modelo de clasificación
<b>Almacenamiento</b>	
Sonómetro	32 GB o 16 GB
<b>Respaldo</b>	
Cloud	Digital Ocean, Google Drive
<b>Comunicación</b>	Remota
<b>Energía</b>	
Continua	5.0 VDC, 3 A
<b>Dimensiones</b>	
Sonómetro	9cm x 6cm x 3cm
Contenedor	22cm x 17cm x 8 cm
<b>Ambiente</b>	
Temperatura ideal	Hasta 70°C

Figura 4.1.1: Especificaciones del sonómetro continuo.

Al iniciar el sonómetro continuo este comienza a registrar y enviar la información a la aplicación web, siguiendo el algoritmo de funcionamiento mencionado en 3.4.2. El montaje del sonómetro con la interfaz de audio USB, el micrófono, la banda ancha móvil y con el contenedor se instaló de la siguiente manera:



**Figura 4.1.2:** Montaje Sonómetro continuo.

De la imagen 4.1.2 podemos observar el contenedor del sonómetro, en 1 tenemos la interfaz de audio USB con el micrófono, en 2 tenemos la Raspberry pi 4, en 3 tenemos el transformador para la corriente y en 4 la banda ancha móvil que está conectada vía USB a la Raspberry pi.



**Figura 4.1.3:** Montaje Sonómetro continuo.

En la figura 4.1.3 se aprecia el mismo montaje que en 4.1.2, con la diferencia del dispositivo de banda ancha móvil, que es un modelo diferente al anterior, pero cumpliendo la misma función.

#### 4.1.2. Homologación.

Como se mencionó en 3.6, la homologación del sonómetro continuo se hace respecto a los valores de  $L_{ASeq}$  del sonómetro de clase 1 [46], comparando segundo a segundo los niveles y sacando un promedio de las diferencias. Aplicando este valor promedio a cada valor de  $L_{ASeq}$  del sonómetro continuo, obtenemos un valor preciso en promedios minutales respecto al sonómetro clase 1, recordando que los sonómetros fabricados para SoundAnalytics se guían por la normativa chilena para mediciones de ruidos [10], que exige mediciones minutales de  $L_{ASeq}$ .

La homologación se efectúa en terreno en diferentes entornos sonoros por periodos

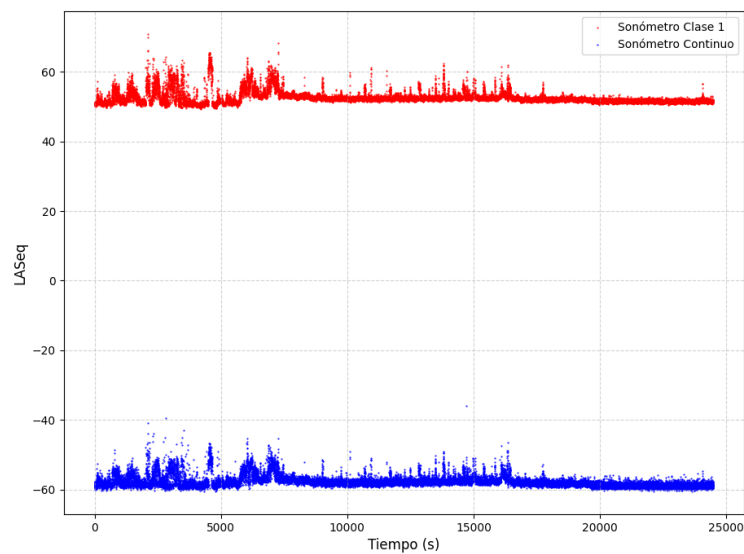
promedios de tiempo de un mínimo de 6 horas.



**Figura 4.1.4:** Medición de niveles para el proceso de homologación.

En la imagen 4.1.4 está el sonómetro clase 1 en un pedestal y el sonómetro continuo en su contenedor registrando el mismo entorno sonoro.

Al momento de analizar y comparar los niveles registrado por ambos dispositivos obtenemos una gráfica de la siguiente manera:

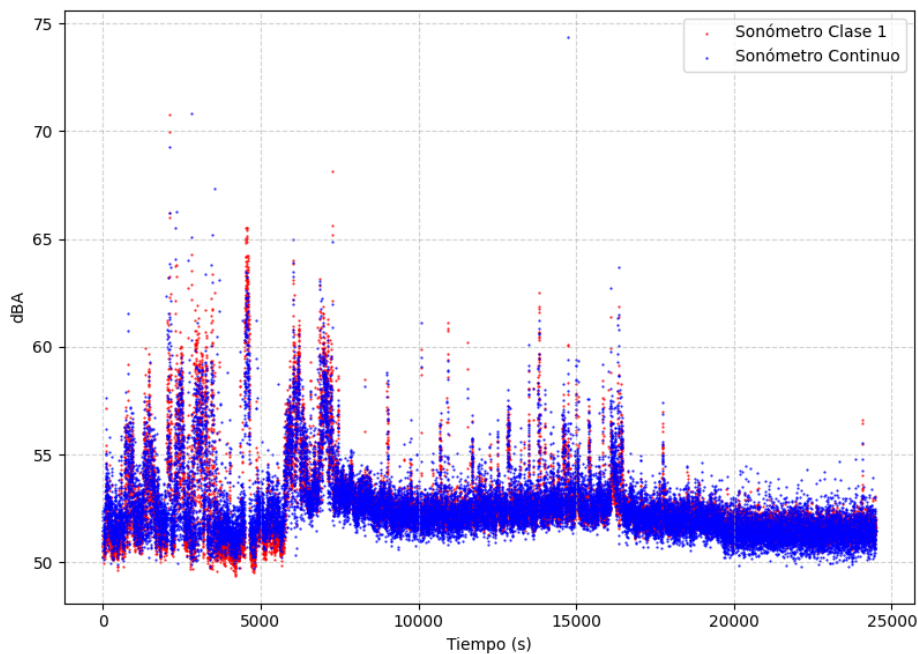


**Figura 4.1.5:** Niveles de  $LASeq$  de ambos sonómetros.



En la figura 4.1.5 vemos los niveles de  $L_{ASeq}$  por segundo para el sonómetro clase 1 y el sonómetro continuo a través del tiempo en segundos, siendo aproximadamente 25000 segundos de datos.

Existe un patrón similar de respuesta de ambos sonómetros diferenciados por un rango de decibels luego del cálculo de la precisión sonora en  $dBA$ , que debemos ajustar para obtener en el sonómetro continuo una respuesta similar al sonómetro de clase 1. Obteniendo el parámetro de homologación y aplicándolo al sonómetro continuo obtenemos:



**Figura 4.1.6:** Aplicación del parámetro de homologación.

Obteniendo así una respuesta muy similar de ambos sonómetros. La norma para los niveles de ruido es en base a promedios minutales, por lo que nos interesa ver el comportamiento de ambos sonómetros en promedios minutales y comparar la efectividad del sonómetro continuo respecto al sonómetro de clase 1. Obteniendo promedios minutales obtenemos:

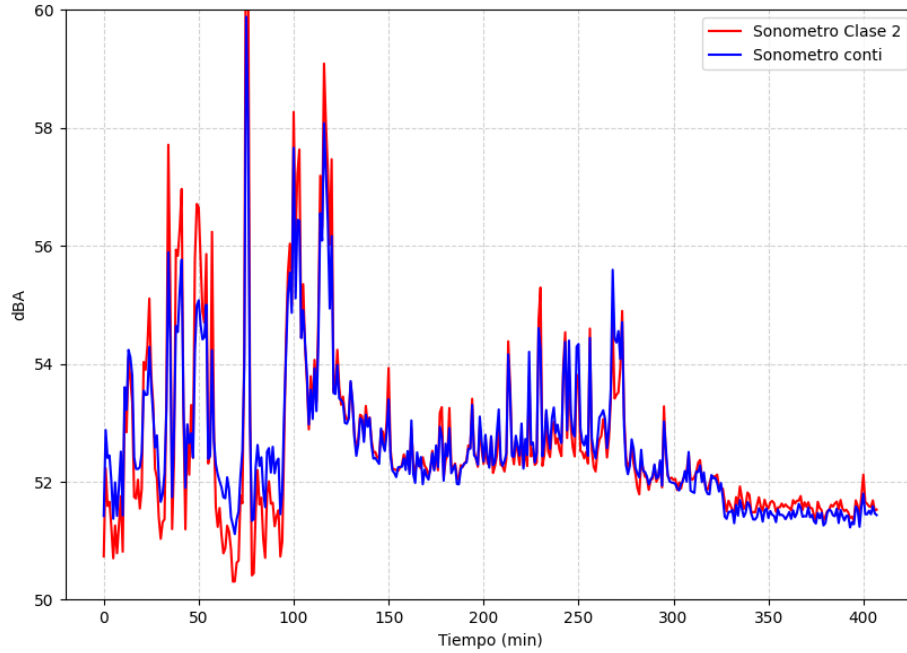


Figura 4.1.7: Promedios minutales de *LASeq*.

Al analizar minuto a minuto la diferencia en los niveles capturados por ambos dispositivos obtenemos que:

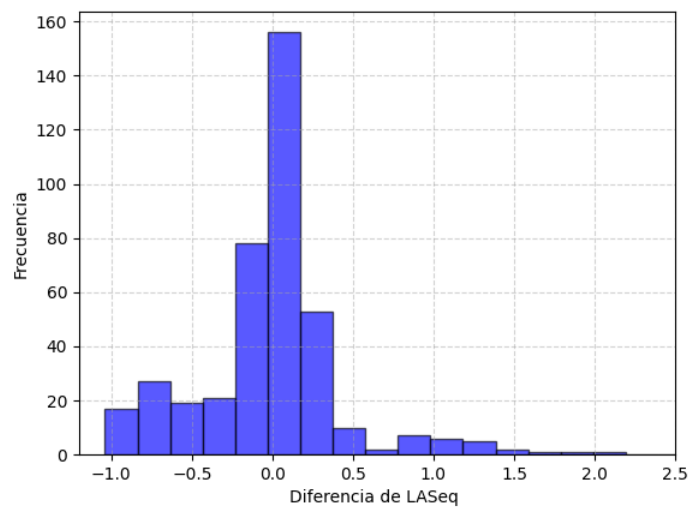


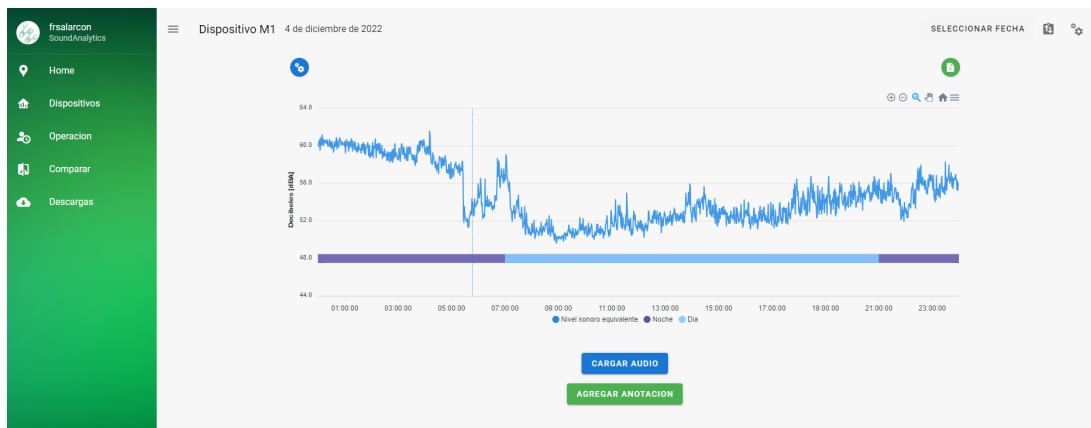
Figura 4.1.8: Frecuencia en diferencias de *LASeq*.

Respecto a los valores de la gráfica 4.1.7 tenemos que el promedio de las diferencias

de niveles por minuto es de  $0,3dB$ , con una desviación estándar de  $0,37$ , siendo la diferencia máxima de  $2dB$  y la mínima de  $0,01dB$ .

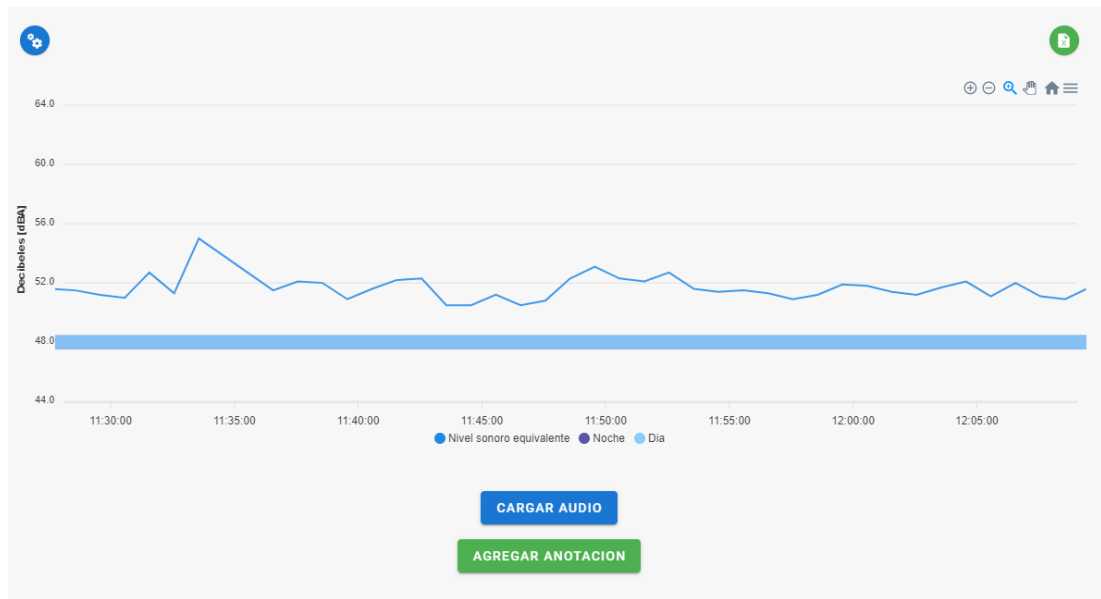
### 4.1.3. Visualización de datos

Los datos capturados por el sonómetro continuo son almacenados en la nube, el promedio minatural de registro de  $L_{ASeq}$  por segundo es enviado a la página de SoundAnalytics, a través de una APIRest [44] para desplegar la información de forma gráfica. Esta gráfica nos mostrará el nivel promedio de  $L_{ASeq}$  por minuto a través del tiempo, dividiendo las gráficas por día.



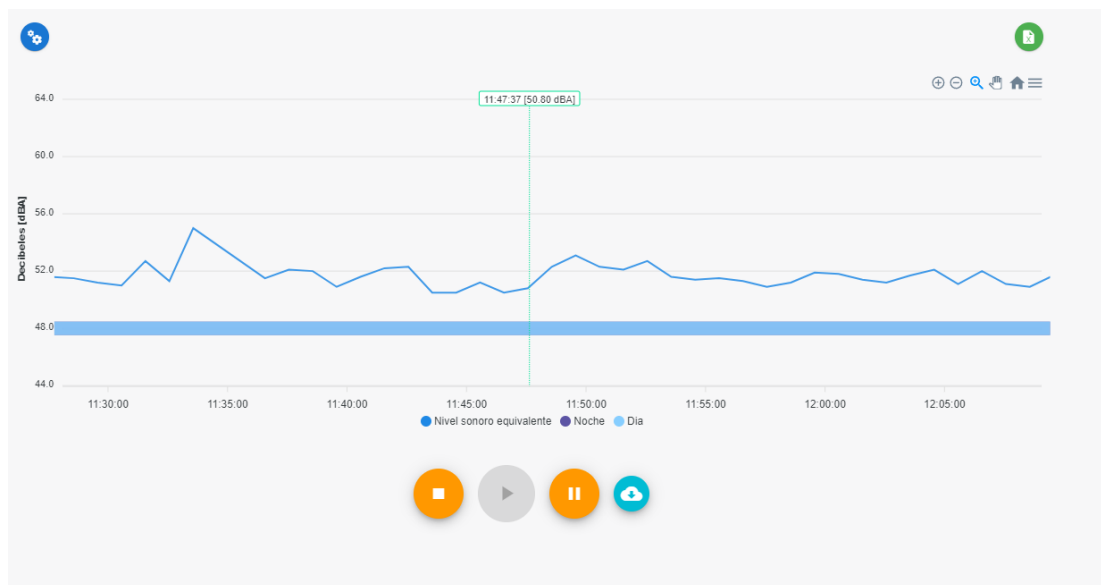
**Figura 4.1.9:** Visualización de los niveles promedio de sonido registrados en la aplicación web de SoundAnalytics.

Vemos en la aplicación web los niveles por minutos en  $dB$  registrados por el sonómetro en tiempo real. La página nos permite aplicar un aumento en las gráficas para obtener mayor detalle de los niveles.



**Figura 4.1.10:** Funcionalidad de acercamiento detallado de los niveles promedio de sonido.

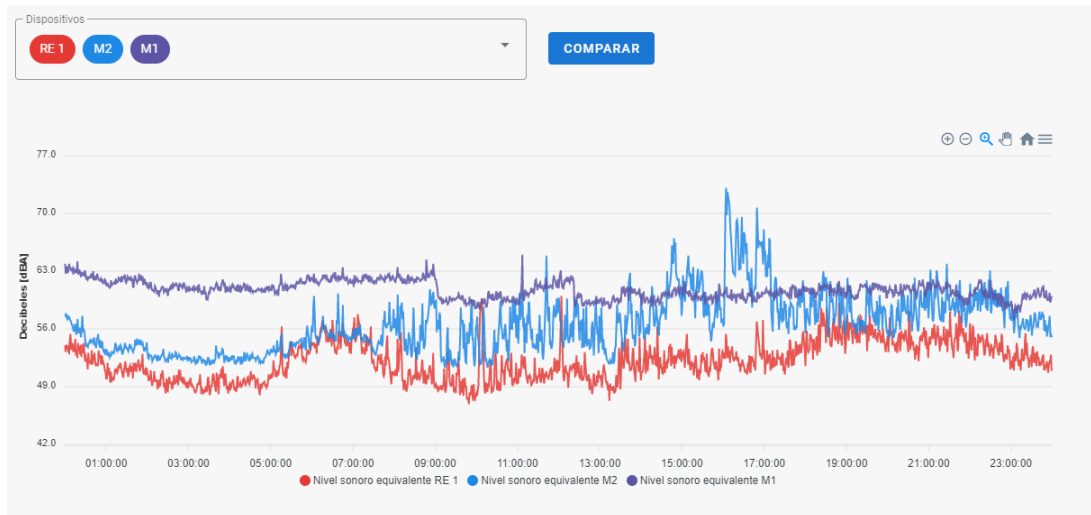
También nos permite seleccionar un minuto en el cual quisiéramos escuchar el audio transcurrido durante ese periodo de tiempo y poder descargarlo.



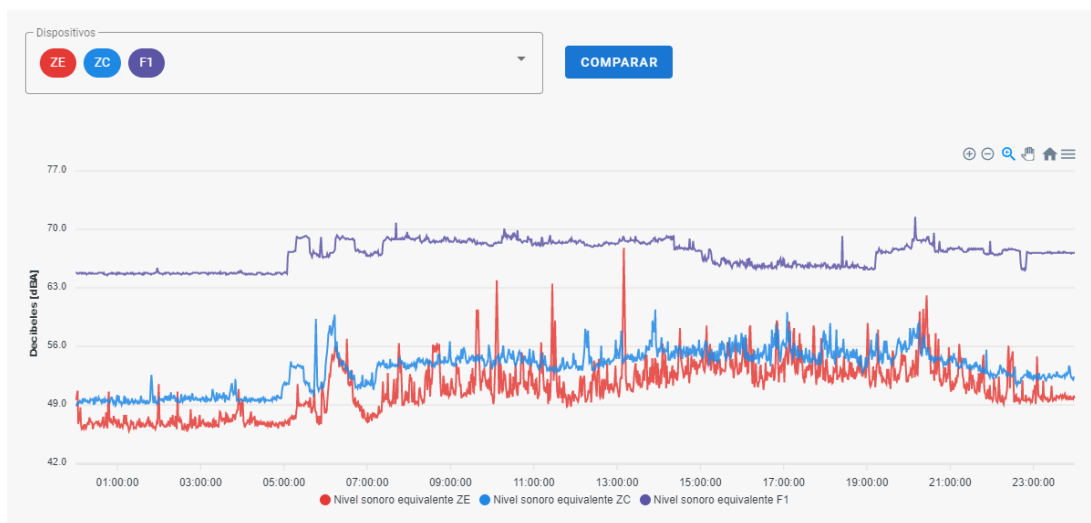
**Figura 4.1.11:** Funcionalidad de reproducción de audios registrados.

Estos audios son extraídos del respaldo que se ejecuta en Digital Ocean [45]. Podemos también crear la comparativa entre los distintos sonómetros funcionando

bajo un entorno sonoro pero en diferentes posiciones de interés.

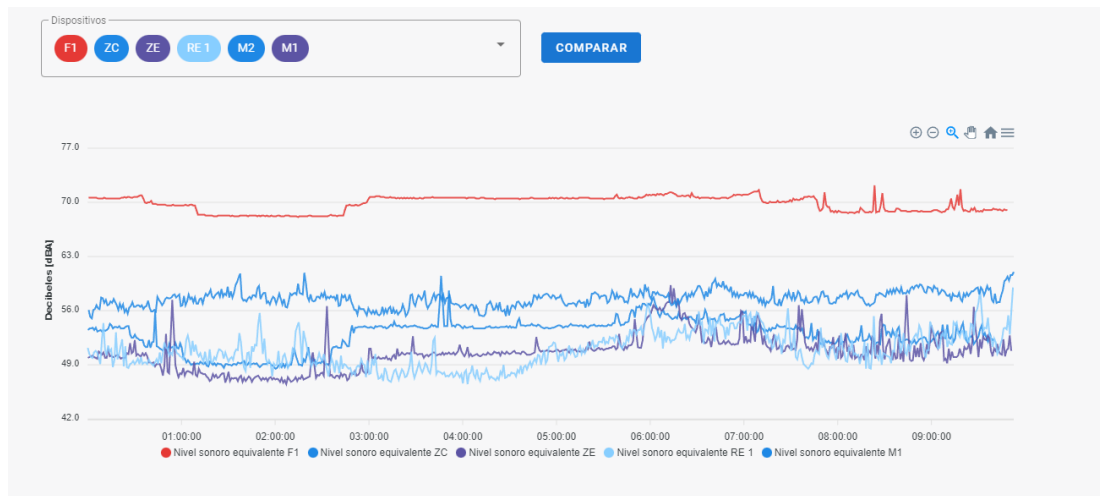


**Figura 4.1.12:** Comparación de niveles de sonido en el entorno sonoro 1.



**Figura 4.1.13:** Comparación de niveles de sonido en el entorno sonoro 2.

Así desplegando en dos entornos sonoros diferentes 3 sonómetros respectivamente.



**Figura 4.1.14:** Total de sonómetros registrando datos en la aplicación web de SoundAnalytics.

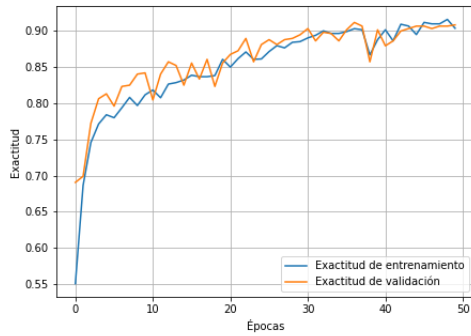
#### 4.1.4. Modelo de Aprendizaje profundo.

Se entrenó el modelo de aprendizaje profundo 3.5.4 con 8 categorías para un entorno sonoro particular, donde interactúan ruido de empresa, ruido de carretera, fauna y población, usando la arquitectura descrita en 3.5.4 y representada en la figura 3.5.3, durante diferentes números de ciclos de entrenamiento. Con un total de 2798 muestras de audios distribuidas en las 8 categorías, con 1958 muestras para el entrenamiento, 588 para el testeo y 252 para la validación, teniendo un total de 176.696 parámetros entrenables.

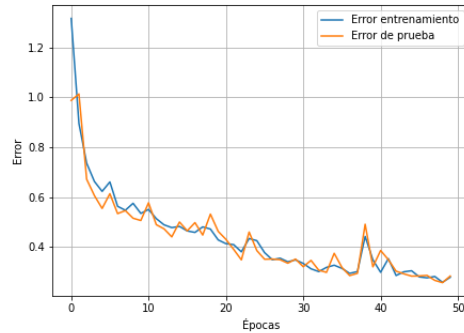
Los resultados de exactitud de entrenamiento y prueba para diferentes ciclos de entrenamiento, resultó en:

Modelo	Número de épocas	Exactitud de entrenamiento	Exactitud de prueba
M1	50	90.1 %	86.9 %
M2	70	92.4 %	91.4 %
M3	100	95.5 %	92.9 %
M4	150	96.5 %	91.1 %
M5	200	99.4 %	93.3 %

Para cada modelo, tenemos que la gráfica de error y exactitud, a través de las épocas de entrenamiento, quedan de la forma:

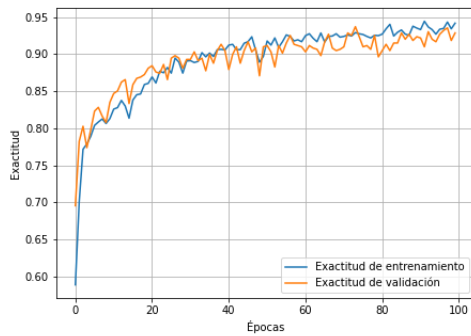


(a) Exactitud de entrenamiento y validación.

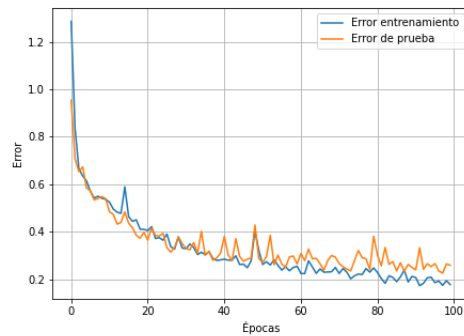


(b) Error de entrenamiento y prueba.

**Figura 4.1.15:** Modelo M1. Entrenamiento con 50 épocas.

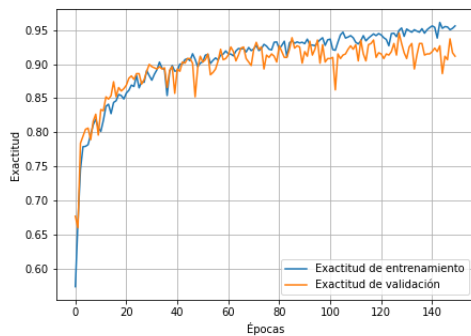


(a) Exactitud de entrenamiento y validación.

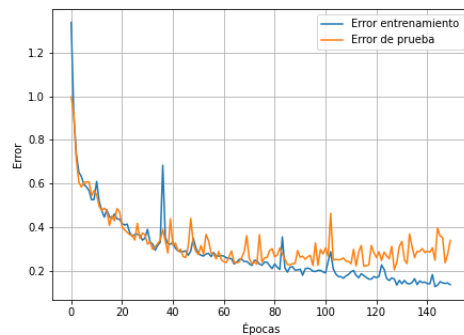


(b) Error de entrenamiento y prueba.

**Figura 4.1.16:** Modelo M2. Entrenamiento con 100 épocas.

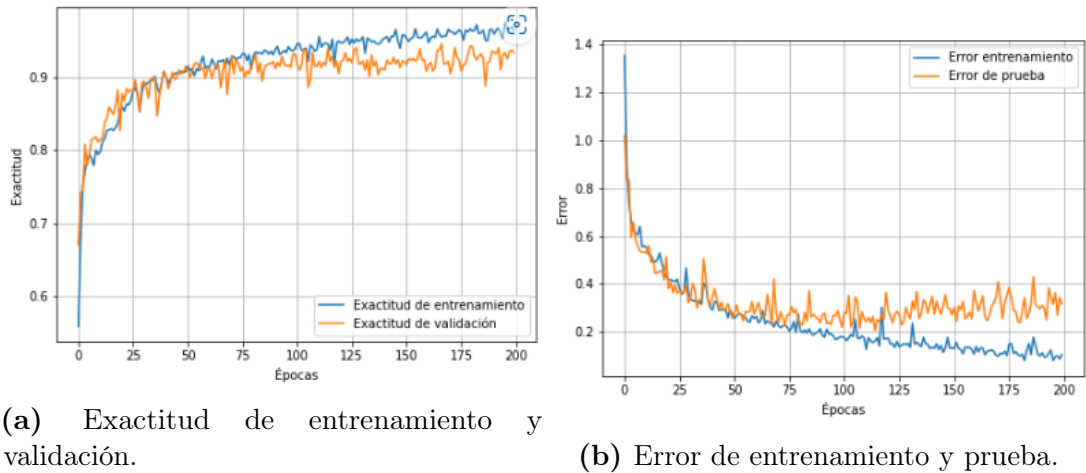


(a) Exactitud de entrenamiento y validación.



(b) Error de entrenamiento y prueba.

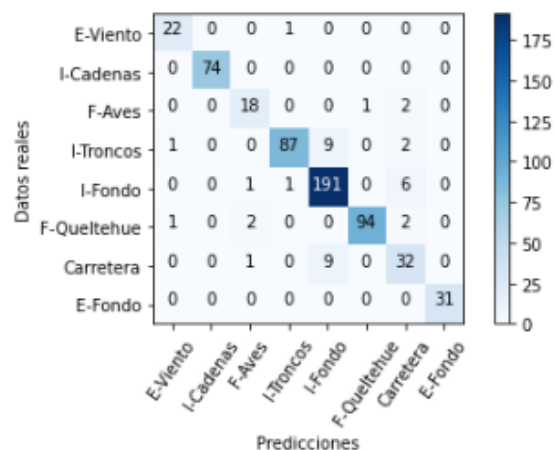
**Figura 4.1.17:** Modelo M3. Entrenamiento con 150 épocas.



**Figura 4.1.18:** Modelo M4. Entrenamiento con 200 épocas.

Donde podemos apreciar que luego de las 100 épocas de entrenamiento 4.1.16 en los gráficos del error respecto a las épocas, comienza a suceder el proceso del sobreentrenamiento, por lo que el modelo M3 fue el elegido para ejecutarse para la muestra de datos en la página, siendo este óptimo y que obtiene mejores resultados de clasificación.

El modelo M3 tiene un comportamiento ideal para la detección de eventos segundo a segundo, podemos ver, a través de una matriz de confusión, la que es una herramienta que permite la visualización del desempeño de un algoritmo, el comportamiento del modelo con los datos de validación.



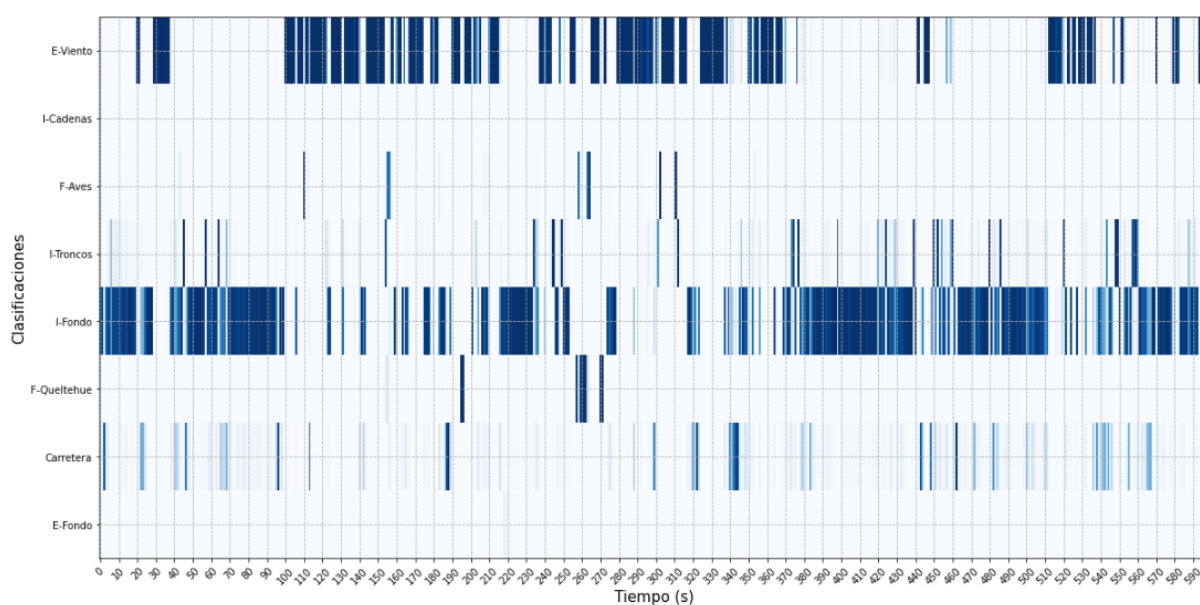
**Figura 4.1.19:** Matriz de confusión del modelo M3.

Donde vemos la efectividad en las predicciones hechas por el modelo, evidenciando que la mayoría de los datos que clasifica, lo hace de manera correcta, mostrando



que no existe mayor confusión entre las clasificaciones realizadas.

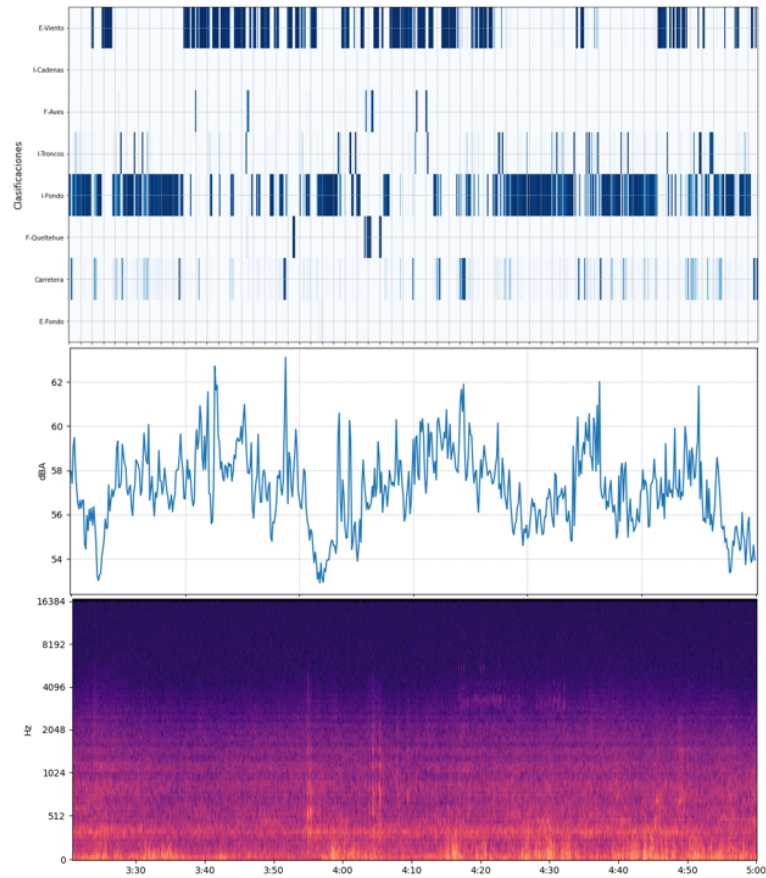
Al aplicar el modelo a un registro de audio aleatorio capturado por el sonómetro, podemos mostrar su comportamiento por segundo, para poder identificar qué elementos del entorno sonoro interactúan:



**Figura 4.1.20:** Ejecución del modelo M3 por segundo.

En la figura 4.1.20 vemos el valor de exactitud de todas las clasificaciones por segundo de forma gráfica, donde las líneas claras son clasificaciones de baja probabilidad y las líneas oscuras son de alta probabilidad. Este tipo de gráfica nos da un contexto de lo ocurrido en el entorno sonoro sin necesidad de escucharlo, así podemos asociar los niveles de presión sonora a lo que detecta el modelo.

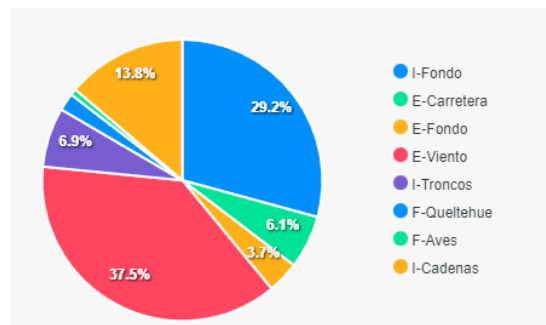
El modelo M3 nos permite ahorrar y optimizar tiempos de análisis para entender los eventos en un entorno sonoro, en un rango de tiempo de interés. La ejecución optimizada del modelo nos permite obtener las clasificaciones de un registro de audio de 10 minutos en 3 segundos en promedio, pudiendo obtener clasificaciones de un día completo en 7 minutos.



**Figura 4.1.21:** Ejecución del modelo M3 por segundo, forma de onda y espectrograma del registro.

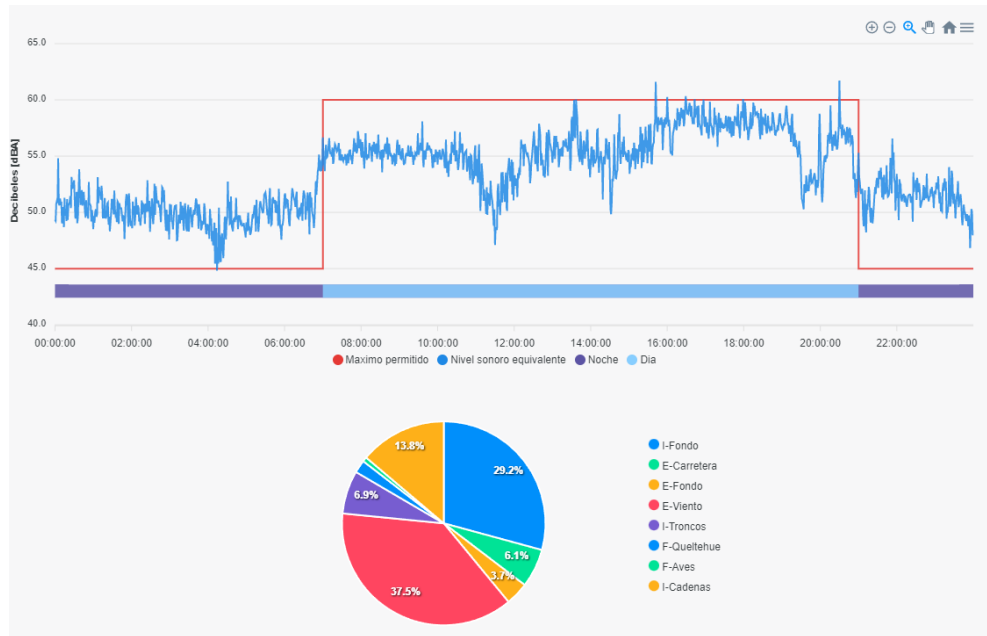
#### 4.1.4.1. Visualización del modelo

En la página el modelo se colocó junto a la gráfica de niveles, en un gráfico el que contiene la estadística diaria de las predicciones hechas segundo a segundo por el modelo, de la forma:

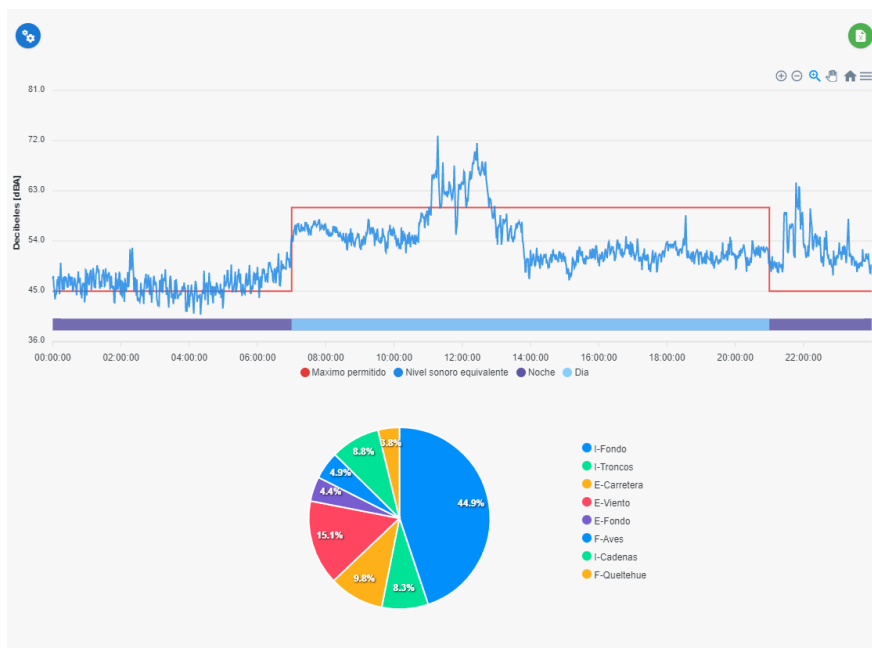


**Figura 4.1.22:** Gráfico de estadísticas diaria de clasificaciones realizadas por el modelo M3.

Ubicado bajo la gráfica de niveles, donde las líneas rojas indican el máximo de  $dBA$  permitido por la norma para el día y la noche [10] respecto a los valores de la tabla 3.0.1 en la zona II.



**Figura 4.1.23:** Visualización de gráfico de estadísticas diaria de clasificaciones realizadas por el modelo M3 en la aplicación web de SoundAnalytics.



**Figura 4.1.24:** Otro ejemplo de visualización de gráfico de estadísticas diaria de clasificaciones realizadas por el modelo M3.

## 4.2. Sonómetro de mano.

El sonómetro de mano quedo con las siguientes especificaciones:

Sonómetro de mano	
<b>Mediciones</b>	
Filtros de medición (dB SPL)	A Z
Tipos de mediciones	SPL, LZSeq, LASeq, LZSeq Max, LZSeq Min, LASeq Max, LASeq Min, Eventos, LZSeq promedio, LASeq promedio, Audio de registro, Audios de eventos
<b>Capacidad de almacenamiento</b>	32 GB
<b>Comunicación</b>	Pendrive vía USB
<b>Display</b>	Pantalla Táctil 3,5" 480x320 píxeles
<b>Energía</b>	
Externa	5.0 VDC, 3 A
Batería externa	300000 mAh
Duración Batería	12 aproximadas
<b>Dimensiones</b>	9cm x 6cm x 3cm
<b>Ambiente</b>	
Temperatura ideal	Hasta 70°C

**Figura 4.2.1:** Especificaciones del sonómetro de mano.

Al implementar todas las funcionalidades descritas en método 3.4 con el hardware 3.4.1 se obtuvo un sonómetro de mano el que tiene las especificaciones de la tabla 4.2.1.

La capacidad de almacenamiento de 32 GB mediante una tarjeta microSD, nos permite registrar hasta un mes de grabación, es decir, 720 horas aproximadamente, debido a la compresión y optimización en los formatos de salida y de almacenamiento. Tiene una duración de 12 horas de uso con una batería externa de 30000 mAh. Al prender el sonómetro y abrir el programa de medición presionando un icono en la pantalla táctil, se nos despliega una interfaz gráfica con los datos que comenzara a registrar. La interfaz gráfica de la aplicación cuenta con 3 pestañas, cuya pestaña 1 se muestra niveles de presión sonora y gráficas, en la pestaña 2

tenemos información más detallada de los niveles y por último en la pestaña 3 se muestra la ejecución del modelo de predicción y clasificación de audio.

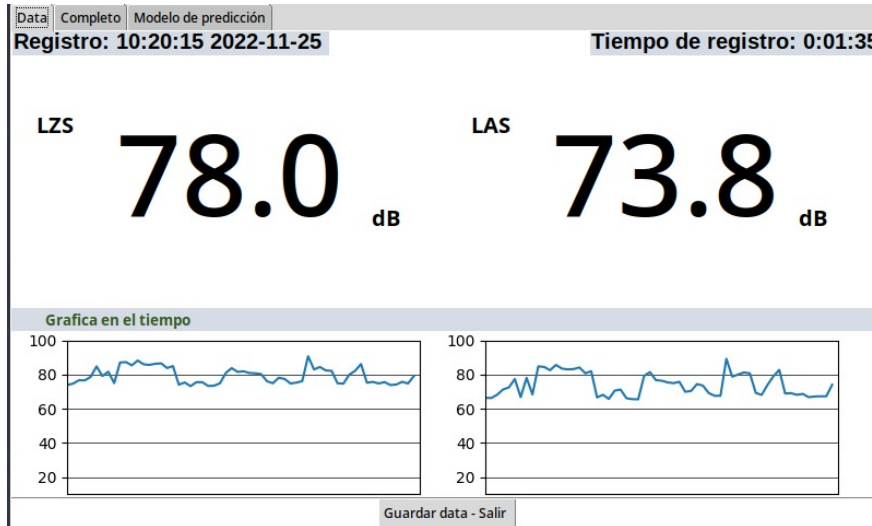


Figura 4.2.2: Data, Pestaña 1 de interfaz gráfica.

En la parte superior de la interfaz 4.2.2 hay 3 botones que direccionan a cada una de las 3 pestañas de la aplicación. Podemos ver en la primera pestaña "Data", información de *LZSeq*, *LASeq* y sus respectivas gráficas segundo a segundo. En la parte superior izquierda esta la fecha y hora del comienzo del registro y en la parte superior derecha el tiempo de registro.

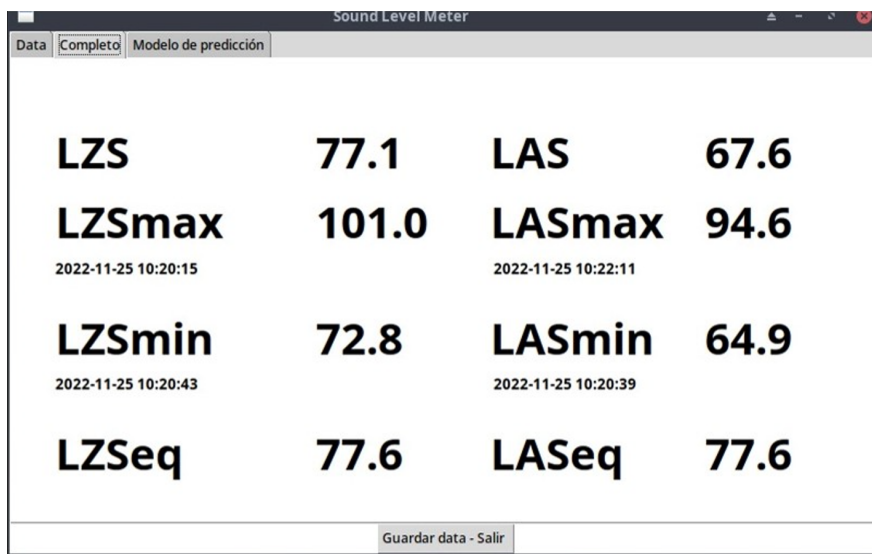
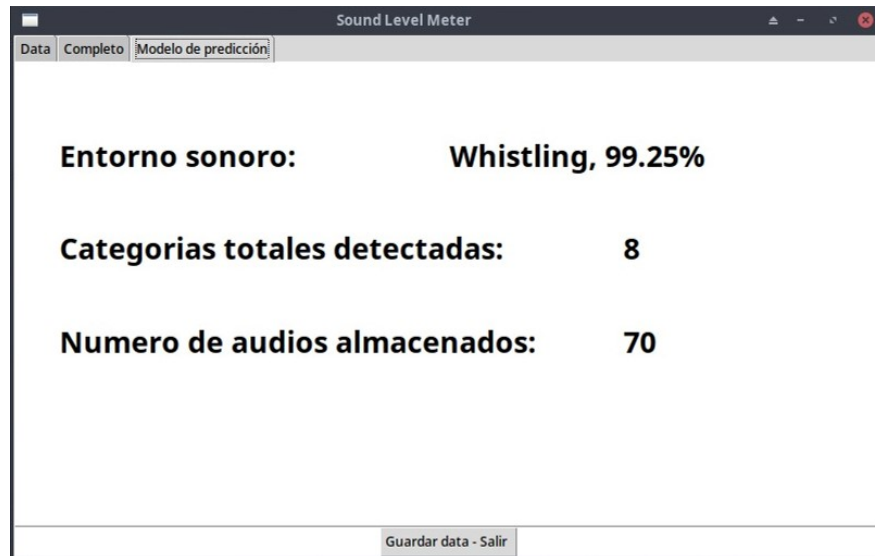


Figura 4.2.3: Pestaña 2 de interfaz gráfica.

En la pestaña 2 se observa la información completa que captura el sonómetro

respecto a los niveles de presión sonora. La parte izquierda de la interfaz indica las mediciones de *LZS* segundo a segundo, el nivel máximo registrado asociado a su fecha y hora de captura, su nivel mínimo y *LZSeq* el nivel promedio de todo el registro el que se actualiza segundo a segundo. De manera análoga, para la parte derecha tenemos los mismos datos pero para *LAS*.



**Figura 4.2.4:** Pestaña 3 de interfaz gráfica.

Por último, la pestaña número 3 de la aplicación nos muestra la ejecución del modelo de clasificación de audios. En la primera fila se presenta una clasificación hecha segundo a segundo con su porcentaje de exactitud asociado, luego en la segunda fila nos indica la cantidad de categorías detectadas (de un total de 521 categorías) que tengan una exactitud mayor al 70% y por último la tercera fila nos indica el número de audios almacenados dentro de esas categorías detectadas. El ensamblaje del sonómetro queda de la siguiente manera:



**Figura 4.2.5:** Sonómetro de mano.

Al conectar a una fuente de energía el sonómetro de mano, con la interfaz de audio USB y micrófono conectados, inicia una pantalla como se aprecia en la imagen de la figura 4.2.5, donde existen 2 aplicaciones para seleccionar. La aplicación en la parte superior es para el registro de audio y la aplicación en la parte inferior es para la extracción de datos.

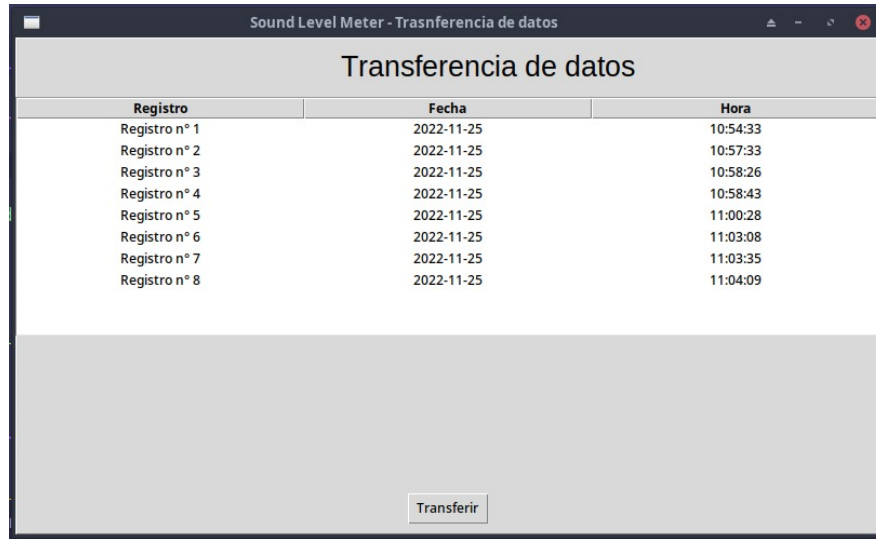


**Figura 4.2.6:** Aplicación de registro de audio.

Al iniciar la aplicación de registro de audio, se despliega la interfaz en la pantalla

y comienza a registrar de forma automática. Para terminar con el registro solo basta presionar el botón “Guardar data - Salir” y la aplicación se cierra de forma automática.

El programa de extracción de datos, nos indica una lista de los datos almacenados, su fecha de registro y hora de inicio.



The screenshot shows a software window titled "Sound Level Meter - Transferencia de datos". The window contains a table with three columns: "Registro", "Fecha", and "Hora". The table lists eight records from "Registro n° 1" to "Registro n° 8", all dated "2022-11-25" with various times. A "Transferir" button is located at the bottom center of the window.

Registro	Fecha	Hora
Registro n° 1	2022-11-25	10:54:33
Registro n° 2	2022-11-25	10:57:33
Registro n° 3	2022-11-25	10:58:26
Registro n° 4	2022-11-25	10:58:43
Registro n° 5	2022-11-25	11:00:28
Registro n° 6	2022-11-25	11:03:08
Registro n° 7	2022-11-25	11:03:35
Registro n° 8	2022-11-25	11:04:09



**Figura 4.2.7:** Aplicación de extracción de datos.

Al colocar un pendrive en el sonómetro, este detectará y montará de forma automática el dispositivo. Luego al apretar el botón "Transferir", las bases de



datos, los archivos XLSX (Excel), los archivos de audio, y los archivos de audios clasificados serán enviados al dispositivo, emitiendo una alerta al finalizar la operación.

Los datos se almacenan en una base de datos. Mediante un programa de visualización de base de datos, posterior a un registro de audio, la base de datos queda de la siguiente manera:

fecha	hora	dBZ	dBA	Predict	Exactitud	1/3 LZSeq 6.3	1/3 LZSeq 8	1/3 LZSeq 10	1/3 LZSeq 12.5	1/3 LZSeq 16	1/3 LZSeq 20	1/3 LZSeq 25	1/3 LZSeq 31.5	1/3 LZSeq 40	1/3 LZSeq 50	1/3 LZSeq 63	1/3 LZSeq 80	1/3 LZSeq 100	
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	
1	2022-11-25	10:54:33	82.69	70.13	Animal	40.1	73.2	72.25	73.81	72.02	72.42	72.43	69.33	65.9	67.54	68.32	68.0	66.86	66.34
2	2022-11-25	10:54:34	76.42	63.81	Sliding door	11.81	65.17	60.58	61.98	62.04	65.0	67.0	65.39	63.22	63.53	64.47	66.07	62.4	60.2
3	2022-11-25	10:54:35	76.05	61.91	Music	11.03	58.44	54.72	59.79	61.93	64.29	67.4	64.79	62.34	61.78	64.22	68.88	64.6	58.41
4	2022-11-25	10:54:36	75.99	61.62	Mechanisms	14.16	64.7	60.14	64.76	59.61	62.74	65.86	65.12	61.85	63.29	63.19	66.92	63.61	59.07
5	2022-11-25	10:54:37	100.83	84.9	Whistling	90.98	84.96	88.37	92.05	93.27	91.3	91.58	91.38	87.51	87.22	86.72	86.36	84.39	82.56
6	2022-11-25	10:54:38	93.81	79.91	Beep, bleep	14.48	81.81	82.03	81.35	81.55	81.94	84.99	85.65	83.28	83.25	82.33	81.01	77.17	73.88
7	2022-11-25	10:54:39	80.53	69.34	Inside, small room	16.71	71.83	63.71	68.55	68.05	69.98	70.63	69.47	66.65	66.19	65.89	66.48	65.62	65.16
8	2022-11-25	10:54:40	85.62	77.67	Knock	27.52	71.72	69.13	69.59	70.16	73.43	75.75	76.26	76.09	76.87	74.91	73.54	67.93	65.68
9	2022-11-25	10:54:41	87.76	81.46	Inside, small room	10.34	76.34	70.68	71.64	73.92	76.19	76.03	77.54	77.17	78.11	77.35	73.86	70.18	68.64
10	2022-11-25	10:54:42	86.47	78.59	Inside, small room	27.73	71.5	68.95	70.3	72.78	74.74	76.45	76.28	75.64	77.03	76.03	74.53	71.85	66.69
11	2022-11-25	10:54:43	86.8	84.92	Hands	31.64	72.39	70.85	71.95	70.54	68.71	69.93	67.8	68.33	70.05	71.19	71.56	69.56	67.78
12	2022-11-25	10:54:44	86.08	85.73	Finger snapping	99.2	73.08	67.0	67.72	67.18	65.72	67.65	68.61	66.28	66.41	67.24	67.42	66.95	63.18
13	2022-11-25	10:54:45	84.51	82.28	Cap gun	22.55	70.52	69.13	68.47	64.96	67.74	70.67	69.59	68.91	69.58	70.55	69.68	69.4	68.52
14	2022-11-25	10:54:46	84.28	83.08	Speech	12.11	72.01	65.48	68.72	66.07	68.6	69.53	68.57	66.7	66.19	66.16	66.06	64.38	64.3
15	2022-11-25	10:54:47	91.1	81.65	Whistling	99.41	76.62	71.35	73.44	76.82	80.63	83.05	82.81	80.13	80.93	79.4	79.2	77.11	75.51
16	2022-11-25	10:54:48	89.87	75.09	Whistling	99.22	78.62	75.17	79.77	80.61	82.16	81.28	81.06	77.0	75.94	75.05	74.4	72.36	68.9
17	2022-11-25	10:54:49	80.55	66.61	Whistling	92.72	63.78	67.36	71.03	71.94	72.42	73.89	68.25	61.98	63.66	62.92	67.42	65.26	59.69
18	2022-11-25	10:54:50	92.62	84.17	Speech	40.89	71.1	52.77	63.12	67.75	68.59	72.04	70.86	71.15	71.79	74.74	77.11	78.52	78.41
19	2022-11-25	10:54:51	87.82	75.79	Speech	92.9	77.34	68.14	71.5	72.66	72.09	73.17	72.99	72.26	74.4	73.04	74.21	74.24	73.08
20	2022-11-25	10:54:52	86.52	84.78	Typing	99.66	73.48	70.69	69.44	67.5	67.8	69.68	69.37	68.79	69.31	70.66	70.51	70.85	70.14
21	2022-11-25	10:54:53	82.18	81.44	Typing	84.44	61.48	55.56	61.46	62.55	63.17	65.99	64.1	61.41	64.41	62.55	63.89	62.32	63.71
22	2022-11-25	10:54:54	81.61	78.36	Typing	84.6	65.95	64.02	63.95	65.03	67.58	69.23	68.6	65.49	66.03	68.42	68.56	65.6	65.1
23	2022-11-25	10:54:55	84.87	80.37	Inside, small room	36.67	77.43	70.0	69.03	67.86	69.99	68.47	67.21	67.47	69.08	69.81	68.62	67.52	
24	2022-11-25	10:54:56	87.12	80.61	Whistling	98.33	75.59	75.51	75.56	74.17	75.54	76.05	76.13	76.0	75.75	74.11	72.27	68.76	65.25
25	2022-11-25	10:54:57	85.41	84.49	Whistling	99.24	57.42	59.51	63.79	65.51	68.88	67.62	66.81	64.33	62.85	65.01	68.85	66.27	63.59

Figura 4.2.8: Visualización de base de datos de un registro de audio.

fecha	hora	dBZ	dBA	Predict	Exactitud	1/3 LZSeq 6.3	1/3 LZSeq 8	1/3 LZSeq 10	1/3 LZSeq 12.5	1/3 LZSeq 16	1/3 LZSeq 20	1/3 LZSeq 25	1/3 LZSeq 31.5	1/3 LZSeq 40	1/3 LZSeq 50	1/3 LZSeq 63	1/3 LZSeq 80	1/3 LZSeq 100	
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	
1	2022-11-25	10:54:33	82.69	70.13	Animal	40.1	73.2	72.25	73.81	72.02	72.42	72.43	69.33	65.9	67.54	68.32	68.0	66.86	66.34
2	2022-11-25	10:54:34	76.42	63.81	Sliding door	11.81	65.17	60.58	61.98	62.04	65.0	67.0	65.39	63.22	63.53	64.47	66.07	62.4	60.2
3	2022-11-25	10:54:35	76.05	61.91	Music	11.03	58.44	54.72	59.79	61.93	64.29	67.4	64.79	62.34	61.78	64.22	68.88	64.6	58.41
4	2022-11-25	10:54:36	75.99	61.62	Mechanisms	14.16	64.7	60.14	64.76	59.61	62.74	65.86	65.12	61.85	63.29	63.19	66.92	63.61	59.07
5	2022-11-25	10:54:37	100.83	84.9	Whistling	90.98	84.96	88.37	92.05	93.27	91.3	91.58	91.38	87.51	87.22	86.72	86.36	84.39	82.56
6	2022-11-25	10:54:38	93.81	79.91	Beep, bleep	14.48	81.81	82.03	81.35	81.55	81.94	84.99	85.65	83.28	83.25	82.33	81.01	77.17	73.88
7	2022-11-25	10:54:39	80.53	69.34	Inside, small room	16.71	71.83	63.71	68.55	68.05	69.98	70.63	69.47	66.65	66.19	65.89	66.48	65.62	65.16
8	2022-11-25	10:54:40	85.62	77.67	Knock	27.52	71.72	69.13	69.59	70.16	73.43	75.75	76.26	76.09	76.87	74.91	73.54	67.93	65.68
9	2022-11-25	10:54:41	87.76	81.46	Inside, small room	10.34	76.34	70.68	71.64	73.92	76.19	76.03	77.54	77.17	78.11	77.35	73.86	70.18	68.64
10	2022-11-25	10:54:42	86.47	78.59	Inside, small room	27.73	71.5	68.95	70.3	72.78	74.74	76.45	76.28	75.64	77.03	76.03	74.53	71.85	66.69
11	2022-11-25	10:54:43	86.8	84.92	Hands	31.64	72.39	70.85	71.95	70.54	68.71	69.93	67.8	68.33	70.05	71.19	71.56	69.56	67.78
12	2022-11-25	10:54:44	86.08	85.73	Finger snapping	99.2	73.08	67.0	67.72	67.18	65.72	67.65	68.61	66.28	66.41	67.24	67.42	66.95	63.18
13	2022-11-25	10:54:45	84.51	82.28	Cap gun	22.55	70.52	69.13	68.47	64.96	67.74	70.67	69.59	68.91	69.58	70.55	69.68	69.4	68.52
14	2022-11-25	10:54:46	84.28	83.08	Speech	12.11	72.01	65.48	68.72	66.07	68.6	69.53	68.57	66.7	66.19	66.16	66.06	64.38	64.3
15	2022-11-25	10:54:47	91.1	81.65	Whistling	99.41	76.62	71.35	73.44	76.82	80.63	83.05	82.81	80.13	80.93	79.4	79.2	77.11	75.51
16	2022-11-25	10:54:48	89.87	75.09	Whistling	99.22	78.62	75.17	79.77	80.61	82.16	81.28	81.06	77.0	75.94	75.05	74.4	72.36	68.9
17	2022-11-25	10:54:49	80.55	66.61	Whistling	92.72	63.78	67.36	71.03	71.94	72.42	73.89	68.25	61.98	63.66	62.92	67.42	65.26	59.69
18	2022-11-25	10:54:50	92.62	84.17	Speech	40.89	71.1	52.77	63.12	67.75	68.59	72.04	70.86	71.15	71.79	74.74	77.11	78.52	78.41
19	2022-11-25	10:54:51	87.82	75.79	Speech	92.9	77.34	68.14	71.5	72.66	72.09	73.17	72.99	72.26	74.4	73.04	74.21	74.24	73.08
20	2022-11-25	10:54:52	86.52	84.78	Typing	99.66	73.48	70.69	69.44	67.5	67.8	69.68	69.37	68.79	69.31	70.66	70.51	70.85	70.14
21	2022-11-25	10:54:53	82.18	81.44	Typing	84.44	61.48	55.56	61.46	62.55	63.17	65.99	64.1	61.41	64.41	62.55	63.89	62.32	63.71
22	2022-11-25	10:54:54	81.61	78.36	Typing	84.6	65.95	64.02	63.95	65.03	67.58	69.23	68.6	65.49	66.03	68.42	68.56	65.6	65.1
23	2022-11-25	10:54:55	84.87	80.37	Inside, small room	36.67	77.43	70.0	69.03	67.86	69.99	68.47	67.21	67.47	69.08	69.81	68.62	67.52	
24	2022-11-25	10:54:56	87.12	80.61	Whistling	98.33	75.59	75.51	75.56	74.17	75.54	76.05	76.13	76.0	75.75	74.11	72.27	68.76	65.25
25	2022-11-25	10:54:57	85.41	84.49	Whistling	99.24	57.42	59.											

de octava para *LZSeq* y *LASeq*.

El archivo XLSX que almacena el sonómetro de mano es similar a la base de datos, solo que este agrega un pequeño resumen de datos tales como el promedio total de *LZSeq* y *LASeq*, el valor máximo de *LZSeq* y *LASeq* y el valor mínimo de *LZSeq* y *LASeq*.

Promedio LZSeq	Promedio LASeq	Max LZSeq	Min LZSeq	Max LASeq	Min LASeq								
85,63	77,72	100,23	78,72	95,98	66,92								
Fecha	hora	dBZ	dBa	Predict	Exactitud	1/3 LZSeq 6.3	1/3 LZSeq 8	1/3 LZSeq 10	1/3 LZSeq 12.5	1/3 LZSeq 16	1/3 LZSeq 20	1/3 LZSeq 25	
2022-11-25	11:04:09	80,72	74,84	Speech	20	62,25	60,72	62,54	64,88	69,49	70,53	70,25	
2022-11-25	11:04:10	92,38	84,9	Skateboard	11,19	78,95	77,49	78,31	77,84	77,95	76,5	75,74	
2022-11-25	11:04:11	92,63	84,93	Whistling	99,76	81,98	82,15	81,63	82	82,74	82,04	79,51	
2022-11-25	11:04:12	79,49	73,68	Whistling	98,34	66,62	67,25	68,42	67,72	67,55	68,98	66,75	
2022-11-25	11:04:13	79,06	69,52	Bird	78,95	67,55	65,67	68,59	68,4	69,84	71,19	66,92	
2022-11-25	11:04:14	82,17	71,55	Animal	67,44	72,98	71,86	71,52	72,94	70,24	71,02	68,29	
2022-11-25	11:04:15	87,86	72,26	idle, small roc	10,05	72,36	74,07	75,94	76,78	77,69	79,49	80,06	
2022-11-25	11:04:16	88,91	70,14	Speech	24,39	77,91	79,43	79,53	79,75	77,25	77,22	78,54	
2022-11-25	11:04:17	89,44	81,76	Speech	49,89	67,43	69,8	70,29	76,27	79,56	82,48	80,11	
2022-11-25	11:04:18	88,14	81,09	Speech	29,21	74,14	72,17	73,79	75,45	76,24	76,42	77,69	
2022-11-25	11:04:19	85,89	84,63	Hands	87,1	77,06	69,46	71,97	68,9	69,54	69,37	66,93	
2022-11-25	11:04:20	85,07	84,79	Hands	86,05	65,81	65,26	68,23	65,32	66,73	67,78	64,48	
2022-11-25	11:04:21	85,28	81,84	idle, small roc	45,43	73,36	70,79	72,24	70,36	74,44	75,45	72,77	
2022-11-25	11:04:22	80,05	76,02	Typing	98,36	69,01	62,75	62,04	64,58	68,15	69,55	68,48	
2022-11-25	11:04:23	78,72	74,67	Typing	90,92	62,43	65,73	65,21	63,78	67,8	67,14	66,4	
2022-11-25	11:04:24	85,72	78,3	Typing	42,34	71,71	70,72	72,7	69,01	69,05	68,39	67,89	
2022-11-25	11:04:25	79,8	71,45	Whistling	98,18	71,44	69,93	70,61	66,14	67,53	67,46	66,6	
2022-11-25	11:04:26	85,29	78,18	Hiccup	45,58	67,56	65,88	66,5	64,31	60,18	64,8	62,04	
2022-11-25	11:04:27	89,48	82,81	Fowl	34,67	65,46	61,95	63,89	61,66	61,77	65,33	63,53	
2022-11-25	11:04:28	87,75	77,25	Speech	96,99	69,9	69,99	73,75	69,66	72,84	72,03	70,4	
2022-11-25	11:04:29	81,07	73,91	Speech	29,48	74,17	70,11	70,4	68,03	67,43	69,09	66,26	
2022-11-25	11:04:30	89,74	83,93	Beatboxing	36,07	67,29	66,73	67,78	67,16	64,39	65,33	64,35	
2022-11-25	11:04:31	82,29	66,92	Speech	31,27	71,07	66,73	68,77	71,4	72,22	72,01	72,01	
2022-11-25	11:04:32	79,6	67,79	Speech	62,88	68,12	69,55	70,7	69,07	68,59	68,64	65,55	
2022-11-25	11:04:33	100,23	95,98	Explosion	78,12	85,46	85,62	84,7	85,29	84,86	83	82,68	
2022-11-25	11:04:34	89,56	77,46	Valk, footstep	9,93	80,89	80,01	81,56	79,65	76,56	76,74	75,05	

Figura 4.2.10: Visualización de archivo XLSX del registro de audio.

Los audios de un segundo clasificado por el modelo quedan almacenados en carpetas diferentes según la categoría de clasificación:

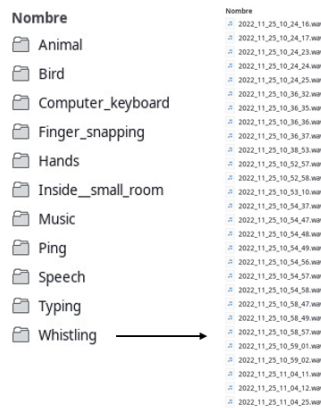


Figura 4.2.11: Carpetas y archivos de audio almacenados por el modelo.

En la figura 4.2.11 vemos carpetas con las clasificaciones detectadas con un 70 % y sus correspondientes archivos de audios, con la fecha y hora de la clasificación. Cada uno de estos archivos de audio son muestras de un segundo, las que nos ayudarán a crear modelos personalizados a futuro armando una base de datos particular bajo un contexto específico de un entorno sonoro.

Para una Raspberry pi 4 de 8 GB de memoria RAM, el rendimiento del dispositivo ejecutando el registro de audio, tiene las siguientes características en promedio:

CPU	RAM	Temperatura	S.O
30 %	7.2 %	35°	Linux Debian 11.5

**Figura 4.2.12:** Valores promedios de rendimiento del sonómetro de mano.

## Capítulo 5

### Discusión

Ambos sonómetros desarrollados cumplieron con una buena medición respecto al registro del sonómetro de clase 1 [46], teniendo 0,3 dB de diferencia en sus mediciones minutales promedios, lo que para entornos sonoros en el contexto de industrias, fauna y población, puede no implicar una gran diferencia, ya que entre eventos de interés, las variaciones en decibeles por lo general suelen ser mucho mayores a 0,3 dB.

Los sonómetros de clase 1, dependiendo la marca y herramientas que este tenga, están en un rango de precio que comienza entre 4 a 5 veces lo que cuesta el equipo completo del sonómetro continuo, siendo el valor aproximado del sonómetro continuo de unos 200,000 a 220,000 pesos chilenos usando el modelo de la Raspberry pi 4 [30] de 8 GB de RAM, el cual es de mayor precio entre sus modelos. Teniendo en cuenta que usamos aproximadamente solo el 30 % de su procesador y cerca del 10 % de su memoria RAM, teniendo así mucho compute para realizar otras tareas y funcionalidades que podemos agregar al sonómetro, se nos presentan dos caminos a seguir:

- Usar un modelo de ordenador de placa simple con menos capacidad para así poder reducir aún más el costo del sonómetro continuo.
- Agregar más funcionalidades al sonómetro.

Debido a que con el modelo Raspberry pi 4 [30] de 8 GB de RAM quedan libres muchos recursos, podemos elegir un modelo más económico y así poder utilizar de manera más óptima la capacidad de procesamiento de otro ordenador de placa simple como pueden ser los modelos de Raspberry 4 de menor memoria RAM

los cuales reducen en gran medida el precio del sonómetro. También se puede continuar con el modelo de Raspberry de 8 GB de RAM y agregar aún más funcionalidades al sonómetro y así poder obtener un producto mucho más sólido y con muchas funcionalidades que un sonómetro convencional no tiene, como lo es la ejecución de modelos de aprendizaje profundo, la sincronización web, comunicación web y el respaldo en la nube. Ambas opciones no son excluyentes, se debe encontrar el balance entre precio, funcionalidades y optimización.

La implementación de modelos de aprendizaje profundo es una herramienta como mucho potencial para los análisis y registros sonoros. Con una cantidad menor de datos, comparada a modelos que usan millones de elementos en sus bases de datos, se obtiene un buen resultado en las clasificaciones con el modelo de aprendizaje profundo M3, siendo este utilizado para recolectar y clasificar más datos para el reentrenamiento, logrando así, un ciclo de trabajo donde el modelo es el encargado de encontrar nuevos datos para mejorar sus resultados. El usar aprendizaje automático ahorra tiempo de análisis y genera una base de datos de audios específicos para diferentes entornos sonoros, así logrando una gran base de datos general para la clasificación y reconocimiento de audio en tiempo real. Por otro lado, la optimización del modelo respecto a la velocidad de ejecución, en conjunto de buenas prácticas de programación es una gran ventaja, logrando que en menos de 10 minutos en promedio podemos tener las clasificaciones de un día completo y la velocidad de ejecución suficiente para poder realizar clasificaciones en tiempo real, lo cual es un buen comienzo para ser un modelo en su primera versión.

Una de las limitaciones que tiene el sonómetro es respecto a la calidad del hardware ya que, para lograr resultados más precisos, se podría usar un micrófono y una interfaz de audio de calidad profesional, como es el micrófono del sonómetro de clase 1, para así aumentar la precisión del registro y tener una respuesta en frecuencia similares a las del sonómetro de clase 1, por otro lado, eso implicaría un costo mayor en el desarrollo del sonómetro. Para aumentar la precisión se pueden implementar nuevas formas y técnicas para la homologación, pero siempre se verá limitada a la calidad de respuesta del micrófono utilizado. Para ambos sonómetros desarrollados se pueden implementar mejoras en los contenedores y carcasas que estos utilizan, para así funcionar de manera óptima bajo las distintas condiciones que puedan presentarse en un entorno sonoro, ya sea por las altas temperaturas o lluvia, implementando sistemas de ventilación y mejoras en

el acceso remoto para el monitoreo y las actualizaciones, evitando manipular el equipo, agregando el uso de baterías externas ante posibles cortes de corriente, aprovechando el uso de energía solar. Respecto a la interfaz gráfica del sonómetro de mano puede esta tener un rediseño más intuitivo y claro respecto a los datos, agregando librerías y lenguajes especializados en la creación de interfaces gráficas. Los sonómetros se desarrollaron en base a la norma ANSI S1.11-2004 [23], la que tiene 18 años de antigüedad. Existen normas más actualizadas en donde se mejoran y optimizan diferentes cálculos para obtener niveles más precisos, siendo estas de alto valor monetario y no de fuente pública.

# Capítulo 6

## Conclusión

Con los resultados obtenidos en el desarrollo de los dos sonómetros, podemos evidenciar la capacidad de registro y precisión que podemos lograr con un ordenador de placa simple respecto a un sonómetro de clase 1. Se logro obtener resultados similares a este instrumento profesional y además agregar nuevas funcionalidades que permiten interactuar y mantener de mejor manera el funcionamiento de sonómetro, así cumpliendo con todos los objetivos propuestos para este trabajo. Se lograron obtener estos resultados con elementos de bajo costo, logrando un dispositivo que reduce diez veces el costo mínimo de un sonómetro de clase 1, siendo accesible para empresas en mediciones de entornos sonoros, facultades, estudiantes y docentes en proyectos e investigaciones. En el transcurso de este trabajo, se desarrollaron 6 sonómetro continuos y un sonómetro de mano para SoundAnalytics [12], los que están en completo funcionamiento siendo usados por largos periodos de tiempo, obteniendo un funcionamiento óptimo y un nivel de producto, para el ámbito profesional.

Existe la posibilidad de mejorar estos dispositivos en ámbito de precisión y optimización, pero sin desviar el objetivo de reducir los costos en instrumentación, por lo que el desafío respecto a los sonómetros es testear nuevos ordenadores de placa simple y llegar al balance entre optimización, precisión y precio. El implementar modelos de aprendizaje automático es una gran ventaja respecto a los sonómetros convencionales, donde se obtienen óptimos resultados en clasificaciones y en tiempo de ejecución, lo que fomenta a profundizar en arquitecturas más complejas para llegar a mejorar los resultados obtenidos. El lograr estos objetivos, podría resultar en iniciar procesos de certificación

profesional para estos dispositivos.

El constante crecimiento de la población provocará un crecimiento en el ruido, contaminante que genera efectos adversos en la salud de las personas y la fauna, en virtud de ello, este trabajo significa un punto de partida para desarrollar nuevos dispositivos, más económicos y que cuentan con herramientas que permiten optimizar los procesos que conllevan la medición de ruido, considerando la precisión y normativas que regulan estos dispositivos, para suplir la necesidad enfática de regularización y registros de diferentes entornos sonoros para mejorar el entorno, la salud de las personas y fauna.



## Bibliografía

- [1] H. Jariwala, H. Syed, M. Pandya, and Y. Gajera, “Noise pollution human health: A review,” 03 2017.
- [2] A. Kattakayam, A. Kurian, T. Lhaden, N. Kennedy, and J. Reddy, “Effects of noise pollution on animals.,” 01 2022.
- [3] L. Radle, “The effect of noise on wildlife: A literature review,” 2007.
- [4] W. H. Organization *et al.*, “Hearing loss due to recreational exposure to loud sounds: a review,” 2015.
- [5] M. Basner, W. Babisch, A. Davis, M. Brink, C. Clark, S. Janssen, and S. Stansfeld, “Auditory and non-auditory effects of noise on health,” *The Lancet*, vol. 383, no. 9925, pp. 1325–1332, 2014.
- [6] A. Mollasadeghi, S. J. Mirmohammadi, A. H. Mehrparvar, M. H. Davari, P. Shokouh, M. Mostaghaci, M. H. Baradaranfar, and M. Bahaloo, “Efficacy of low-level laser therapy in the management of tinnitus due to noise-induced hearing loss: a double-blind randomized clinical trial,” *The Scientific World Journal*, vol. 2013, 2013.
- [7] H. J. Jariwala, H. S. Syed, M. J. Pandya, and Y. M. Gajera, “Noise pollution & human health: a review,” *Indoor Built Environ*, pp. 1–4, 2017.
- [8] T. Münzel, M. Sørensen, and A. Daiber, “Transportation noise pollution and cardiovascular disease,” *Nature Reviews Cardiology*, vol. 18, no. 9, pp. 619–636, 2021.
- [9] Issam, M. Abdelraziq, H. Ali-Shtayeh, and Abdelraziq, “Effects of noise pollution on blood pressure, heart rate and hearing threshold in school children,” *Journal of Applied Sciences*, 2003.
- [10] “Decreto n° 38 del ministerio del medio ambiente, norma de emisión de ruidos molestos,” 2011.
- [11] I. E. Commission, “Electroacoustics - sound level meters,” Standard IEC 61672-2:2003, International Electrotechnical Commission, 2003.
- [12] SoundAnalytics, *SoundAnalytics - Análisis en tiempo real de entornos sonoros*. SoundAnalytics, 2022.

- [13] SocioAnalytics, *SocioAnalytics - Inteligencia Socioambiental*. SocioAnalytics, 2023.
- [14] A. French, *Vibraciones y ondas*. Curso de Física Del M.I.T Series, Editorial Reverté, 1974.
- [15] M. P. Norton and D. G. Karczub, *Sound waves: a review of some fundamentals*, p. 128–192. Cambridge University Press, 2 ed., 2003.
- [16] P. R. Jones, “What’s the quietest sound a human can hear,” *Aka “Why Omega-3 fatty acids might not cure dyslexia*, 2014.
- [17] G. Fechner, “1966. elements of psychophysics,” *Elements of Psychophysic. 1st Ed. New York, NY: Holt, Rinehart, and Winston Inc*, 1860.
- [18] “Ieee standard definitions for selected quantities, units, and related terms, with special attention to the international system of units (si),” *IEEE Std 270-2006 (Revision of IEEE Std 270-1966)*, pp. 1–43, 2006.
- [19] “Acoustics — normal equal-loudness-level contours,” standard, International Organization for Standardization, 2003.
- [20] A. Jaramillo, P. Burke, and D. Beadle, *Aves de Chile: incluye la península Antártica, las islas Malvinas y Georgia del Sur*. Descubrir la naturaleza: Guías, Lynx Edicions, 2005.
- [21] J. W. Cooley and J. W. Tukey, “An algorithm for the machine calculation of complex fourier series,” *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.
- [22] C. Gauss and K. der Wissenschaft, *Carl Friedrich Gauss Werke: Dritter Band*. Salzwasser-Verlag, 2021.
- [23] Acoustical Society of America, “Specification for octave-band and fractional-octave-band analog and digital filters,” Standard S1.11-2004, American National Standards Institute, 2004.
- [24] Y. Suzuki and H. Takeshima, “Equal-loudness-level contours for pure tones,” *The Journal of the Acoustical Society of America*, vol. 116, no. 2, pp. 918–933, 2004.
- [25] R. A. Española, “Aprender,” vol. 23, 2022.
- [26] M. A. Nielsen, *Neural Networks and Deep Learning*. 2015.
- [27] A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, 2017.
- [28] J. Patterson and A. Gibson, *Deep Learning: A Practitioner’s Approach*. O’Reilly, 2017.
- [29] R. P. Foundation, *Raspberry Pi*.

- 
- [30] R. P. Foundation, *Raspberry Pi 4*.
- [31] BOYA, *Microfono BOYA BY-M1*.
- [32] Ventiontech, *Tarjeta de sonido USB VENTION CDLH0*.
- [33] Python Software Foundation, *Python: programming language that lets you work quickly and integrate systems more effectively*. Python Software Foundation, 2022.
- [34] Matthias Geier, *Sounddevice: Module provides bindings for the PortAudio library and a few convenience functions to play and record NumPy arrays containing audio signals.*, 2022.
- [35] PortAudio community., *PortAudio: free, cross-platform, open-source, audio I/O library*.
- [36] Travis Oliphant, *Numpy: The fundamental package for scientific computing with Python*.
- [37] Dwayne Richard Hipp, *SQLite: C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine*.
- [38] TensorFlow, *YAMNet: Pretrained deep net that predicts 521 audio event classes based on the AudioSet-YouTube corpus*.
- [39] Google, *AudioSet: A large-scale dataset of manually annotated audio events*.
- [40] E. Fonseca, M. Plakal, F. Font, D. Ellis, X. Favory, J. Pons, and X. Serra, “General-purpose tagging of freesound audio with audioset labels: Task description, dataset and baseline,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, 2018.
- [41] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *CoRR*, vol. abs/1704.04861, 2017.
- [42] G. Brain, *TensorFlow: free and open-source software library for machine learning and artificial intelligence*.
- [43] G. v. R. Python, Steen Lumholt, *Tkinter: estándar de interfaz gráfica de usuario (GUI) para Python*.
- [44] M. Masse, *REST API Design Rulebook*. O’Reilly Media.
- [45] Digital Ocean, *DigitalOcean: Proveedor estadounidense de servidores virtuales privados*. DigitalOcean, Inc., 2022.
- [46] Larson Davis, PCB division, *SoundTrack® LxT Workplace Noise Sound Level Meter*. Larson Davis, 2022.